Projet Sample Orchestrator
Rapport final

Contributeurs: Tous participants au projet
Edition et validation : Hugues Vinet
Sous-projets: tous
Statut diffusion: Restreinte
Date (mois/année) : Juin 2009
Version: V1

Résumé:

Ce document, constituant le délivrable final du projet Sample Orchestrator, fournit une synthèse des travaux réalisés pour la totalité du projet. Il est organisé en 3 parties : « Programme des actions engagées » décrivant l'ensemble des travaux réalisés, selon la structuration des tâches définies dans l'annexe technique et précisée dans le document de spécification détaillée, « Résultats scientifiques et techniques », présentant les principaux résultats issus des travaux de recherche et leur traduction sous la forme de modules logiciels et des deux principales applications réalisées : Echantillonneur et Prototype d'aide à l'orchestration, et « Diffusion et valorisation des travaux », recensant les principaux objets, produits et actions de diffusion et valorisation scientifique, professionnelle et publique liés aux résultats du projet. L'ensemble des participants au projet ont contribué à sa rédaction.

Table

1 Programme des actions engagées

1.1 Description générale du projet

1.1.1 Présentation synthétique

Le projet Sample Orchestrator vise la réalisation et l'expérimentation de nouvelles applications reposant sur la gestion et la manipulation par le contenu de banques d'échantillons sonores (sons isolés pré-enregistrés).

Sa réalisation s'inscrit à la conjonction de deux processus : d'une part la disponibilité commerciale de grandes banques d'échantillons diffusées sur différents supports (CD et DVD, bases en ligne), mais limitées dans leurs applications (synthétiseurs par échantillonnage) ; d'autre part, des avancées scientifiques et technologiques récentes en matière de méthodes d'indexation et de systèmes de gestion de bases de données audio, permettant d'envisager des fonctions musicales encore inédites, faisant appel à des **modes de gestion globale et de manipulation par le contenu** de l'ensemble des échantillons disponibles :

- *gestion par le contenu de banques d'échantillons* : l'objectif est ici de dépasser les systèmes de classement existants, généralement fondés sur des taxonomies figées, en autorisant d'une part des modes de classification personnalisées par les utilisateurs en fonction de leurs propres catégories et corpus et définis à partir d'exemples caractéristiques de chaque catégorie, d'autre part de permettre des recherches par similarité, de manière transversale à tous ces systèmes de classification, se fondant uniquement sur certains critères acoustiques choisis par l'utilisateur ; le développement de telles fonctions s'impose avec la multiplication du nombre de sons disponibles et la nécessité de définir de nouvelles heuristiques pour trouver des sons sur la base de leurs caractéristiques sonores ;
- *traitement audio par le contenu* (se fondant sur une analyse préalable des caractéristiques des signaux traités) : transposition, dilatation, filtrage, mise en boucle, édition temps-fréquence, etc. Ces fonctions de traitement se décomposent en fonctions de pré-traitement, visant la préparation d'échantillons en studio, et celles de post-traitement, intervenant en temps réel lors de l'utilisation de banques de sons par un dispositif instrumental d'échantillonneur de nouvelle génération ; l'objectif est ici de constituer de nouveaux instruments de musique, combinant sons pré-enregistrés, caractéristiques pré-calculées de ces sons et traitements temps réel modulés par le jeu et par ces caractéristiques ;
- *outil d'aide à l'orchestration*, trouvant, sur la base de la modélisation de connaissances musicales, généralement empiriques de la part de compositeurs, les meilleures combinaisons de sons en vue de l'obtention de l'effet musical recherché (fusion/ séparation, ré-orchestration à partir de sons existants, etc.).

Le projet est rendu possible à travers la collaboration de deux partenaires :

- l'Ircam (coordinateur), centre de recherche à la pointe des travaux au niveau international en indexation et traitement du signal audio et outils de composition avancés, mais aussi lieu d'accueil et de collaboration avec de nombreux compositeurs du monde entier, et de développement et de diffusion de logiciels musicaux de pointe dans le cadre du Forum Ircam (1500 utilisateurs, essentiellement musiciens professionnels, dans le monde) ;
- La société Univers Sons, éditeur de logiciels et bases de données orientées sons et instruments, qui est le leader et pionnier dans le domaine des instruments virtuels à base de d'échantillonnage avancé. La société a conçu le moteur UVI Engine,

technologie licenciée par les plus grandes sociétés spécialisées dans la manufacture d'instruments logiciels novateurs, et utilisée par des dizaines de milliers de musiciens, producteurs, sound designers, ingénieurs du sons, qu'ils soient professionnels ou amateurs éclairés.

L'organisation du projet prévoit des activités de recherche autour des principaux verrous scientifiques et technologiques identifiés : description, indexation, et classification automatique des contenus sonores et musicaux, traitement sonore par le contenu, ingénierie des connaissances musicales et leur application à l'orchestration, environnements logiciels pour le traitement audio temps réel.

Ces travaux de recherche sont appliqués au développement de deux prototypes applicatifs : **l'Echantillonneur**, développé par Univers Sons, et un **Outil d'aide à l'orchestration,** développé par l'Ircam dans le cadre de son environnement d'aide à la composition OpenMusic. Ces deux applications reposent sur la gestion des mêmes contenus (bases d'échantillons commerciales) et en partie sur des fondements technologiques communs (système de gestion de données et métadonnées musicales, bibliothèques d'indexation audio, …). Un aspect important du projet réside également dans la mise en œuvre de sessions de retour d'usage à partir de versions successives des logiciels, afin d'adapter le développement des fonctions aux attentes des utilisateurs visés.

Les applications développées s'adressent à un large panel d'utilisateurs (production musicale, post-production son, production multimédia et de jeux, design sonore, composition musicale, etc.) en leur faisant bénéficier de gains de productivité (efficacité, personnalisation, encapsulation de connaissances spécifiques), et de nouveaux potentiels de créativité.

1.1.2 Objectifs scientifiques et industriels

1.1.2.1 Objectifs scientifiques

Le projet intègre des objectifs scientifiques et technologiques relevant de compétences pluridisciplinaires et dépassant l'état de l'art actuel :

- formalisation de descriptions de haut niveau pour les sons isolés, et en particulier les sons non-instrumentaux (cognition sonore et musicale, psychologie expérimentale, ingénierie des connaissances musicales),
- extraction automatique de différentes formes de description de corpus sonores à partir des signaux (indexation et analyse des signaux audionumériques, apprentissage automatique) :
  - classification automatique par apprentissage de catégories définies par l'utilisateur à partir d'exemples de sons prototypes ;
  - extraction de descripteurs caractéristiques des échantillons en vue de la mise en œuvre de mesures de similarité combinant de multiples critères, destinés à la réalisation de fonctions de recherche d'échantillons par l'exemple ;
- nouveaux traitements audio par le contenu (analyse et traitement du signal audionumérique) ;
- application de méthodes de synthèse concaténative à des corpus non-vocaux et mise en œuvre sous la forme d'une interface interactive (traitement du signal audionumérique, interfaces homme-machine) ;
- conception et réalisation d'un outil d'assistance à l'orchestration (ingénierie des connaissances musicales, langages, algorithmique, systèmes de traitement de données symboliques, interfaces homme-machine) ;

- conception et développement de système de gestion de bases de données et métadonnées et de middleware associé (ingénierie informatique) ;
- conception et développement de moteurs de synthèse et de traitement en temps réel des échantillons (ingénierie de systèmes audio)
- spécification et suivi de protocoles d'analyse des besoins et de retour d'usage (ingénierie des besoins et des usages, ergonomie cognitive).

1.1.2.2 Objectifs industriels et applicatifs

S'appuyant sur l'expérience d'acteurs reconnus de la production d'applications logicielles orientées instruments virtuels et reconnaissance temps réel, ce projet vise la réalisation de fonctions musicales inédites, concernant toutes la gestion de banques d'échantillons sonores, et visant des populations intervenant dans différents types d'activités:

- la production musicale : travail en studio et sur scène,
- la composition musicale (aide à la composition),
- la post-production sonore (effets spéciaux à l'image notamment)
- le jeu vidéo (effets et environnements sonores),
- la production multimédia (idem),
- l'illustration sonore
- la gestion de bases de données audio (extra-musicales, dans différents domaines professionnels : simulation, zoologie, etc.).

La banalisation des banques sonores à destination des médias numériques et plus généralement leur transfert du support CDROM et CD Audio initial vers des formats semi propriétaires, a favorisé l'émergence de produits de lecture de ces banques. Le principe de l'instrument virtuel s'avère être le plus adapté bien que souffrant encore de nombreuses lacunes, à la fois dans le mode de gestion, mais aussi à travers l'analyse et l'édition de son contenu.

Dans ce contexte à la pointe des technologies de gestion des médias sonores à base d'échantillons, on retrouve les acteurs marquant de ce secteur (Native Instruments, Spectrasonics, Synful, VSL, IK Multimedia, Mark Of The Unicorn, Symbolic Sound). Le projet Sample Orchestrator vise le développement d'un outil de 3eme génération (Génération 1 : le CD audio/ROM, Génération 2 : Les premiers instruments virtuels de gestion de données audio), comblant le fossé constaté entre les besoins utilisateurs et la technologie disponible sur le marché.

- Le projet **Sample Orchestrator** se destine à plusieurs types d'opérateurs via des outils adaptables et personnalisables selon les pratiques en vigueur au sein de chaque corporation.
- Évolution du mode de gestion orchestral, dépassant le cadre classique des instruments (banques de sons précalculées) pour proposer l'orchestration des échantillons , quelle que soit leur signature sonore, selon de nouvelles méthodes et approches reposant sur une prise en compte *globale* (technologies de bases de données) et *par le contenu sonore* (à travers l'utilisation de descripteurs sonores et musicaux pertinents pour les fonctions visées, automatiquement extraits des signaux audionumériques).
- Importation et exportation multi formats et multi critères

Le consortium réunit des compétences reconnues dans les trois domaines clés nécessaires à la réussite du projet :

- des experts en description et indexation des corpus sonores et musicaux,
- des experts en traitement des signaux audionumériques,
- des experts en ingénierie d'échantillonnage,
- des industriels de premier plan qui maîtrisent et développent des usages et pratiques métiers correspondants aux champs d'applications visés.

Le projet **Sample Orchestrator**, face aux technologies en place, se propose d'aller plus loin, voire d'innover dans les domaines suivant :

- Orchestration musicale, visuelle et sonore: Interaction temps réel entre les représentations graphiques des sons et leur modes de traitement.
- Recherche intelligente à l'intérieur d'un contenu complexe et multiple, par similarité, proximité, empreintes et non plus seulement sur la base de catégories pré-définies
- Traitement de la matière sonore en temps que telle (sons instrumentaux, sons et bruits identifiables ou non)

Le projet vise à renouveler les modes d'appréhension et de travail autour des banques d'échantillons, qu'elles relèvent de modes de classification objectifs (les différentes catégories d'instruments) ou de caractéristiques acoustiques relevant d'environnement sonores quelconques. L'orchestration de ces samples d'origines diverses, associant le mode graphique d'édition, la recherche intelligente de critères et la pertinence d'une interface utilisateur accessible et performante, forment le socle de ce projet dans son cadre d'utilisation potentielle.

Le projet vise la mise en œuvre de tâches de recherche destinées à réaliser ces fonctionnalités nouvelles et leur implémentation sous la forme de deux prototypes applicatifs principaux :

- Une *application d'échantillonneur de nouvelle génération*, développée par Univers sons sur la base d'évolutions de son moteur UVI Engine.Cette application logicielle permettra la gestion et l'édition des médias sonores multi formats ainsi que leur édition temps réel, l'indexation du contenu et sa gestion multi critères. À la fois application autonome (Stand Alone) et insérable (plug in) au sein de l'ensemble des applications hôtes existantes sur le marché, cette application doit répondre à des attentes précises et non résolues des opérateurs des domaines de travail évoqués plus haut.
- Une *application d'outil d'aide à l'orchestration*, développée par l'Ircam et intégrée dans son environnement logiciel d'aide à la composition OpenMusic, et répondant aux besoins tant de compositeurs experts en composition mais en attente d'outils stimulant leur créativité, que de compositeurs non expérimentés s'appuyant sur les connaissances musicales intégrées pour les assister dans leur travail d'orchestration.

Ces recherches et développements font l'objet de confrontations, tant en amont (explicitation et analyse des besoins) qu'en aval (retour d'usage), selon des méthodologies et protocoles formalisés, afin de garantir leur adéquation aux utilisations visées.

1.1.3 Organisation

Structure du consortium

Le projet associe l'IRCAM[1] (coordinateur), laboratoire de recherche et institution de production musicale contemporaine, et l'industriel Univers sons[2]. L'Ircam a été en charge de la coordination du projet et des recherches sur l'analyse et le traitement des signaux sonores (équipe Analyse/synthèse – Ircam-AnaSyn), sur la perception et le design sonores (équipe Perception et design sonores, Ircam-PDS), sur l'informatique musicale, les outils d'aide à la composition et le développement de l'application d'aide à l'orchestration (équipe Représentations musicales, Ircam-RepMus) et sur application musicales interactives (équipe Interactions musicales temps réel, Ircam-IMTR). Univers Sons a été en charge du développement et de l'industrialisation de l'application Echantillonneur intégrant les modules issus des recherches de l'Ircam.

Organisation des tâches

Le projet est organisé selon une division en 5 sous projets (SP1 à SP5), chacun d'eux étant coordonné par une des équipes d'un partenaire, et donnant lieu à des rendus spécifiques (délivrables) et à ses propres critères de validation. Les sous-projets sont les suivants :

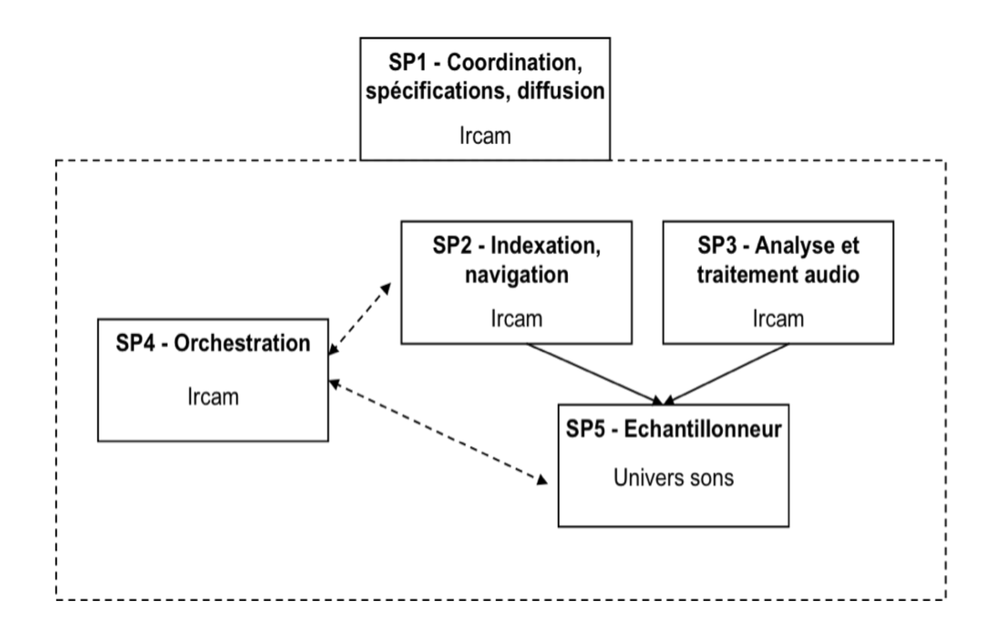
- *SP1- Gestion, spécifications, méthodologie, diffusion*: toutes activités communes liées à la coordination et à la diffusion du projet, ainsi que, en début de projet, une phase de spécifications visant à préciser le programme de travail et les spécifications techniques et fonctionnelles, et la définition des méthodologies, notamment pour les processus de retour d'usage à partir des versions intermédiaires des applications ; Coordination : Ircam ; Participants : tous.
- *SP2-Indexation audio et navigation par le contenu* : recherche et développement pour la formalisation de descriptions musicales et sonores de haut niveau, le calcul de métadonnées musicales par extraction automatique à partir des signaux audionumériques, et la mise en œuvre de ces métadonnées pour la réalisation de fonctions et interfaces de navigation par le contenu ; Coordination : Ircam ; Participants : Ircam-AnaSyn, Ircam-PDS, Ircam-IMTR, Univers Sons.
- *SP3-Analyse et traitement audio* : recherche et développement en vue de la réalisation de fonctions de post-traitement des échantillons sonores, sur la base de descripteurs pré-calculés ; adaptations et évolutions du moteur UVI engine pour l'intégration de ces nouvelles fonctions de traitement ; Coordination : Ircam ; Participants : Ircam-AnaSyn, Ircam-IMTR, Univers Sons.
- *SP4-Orchestration* : recherche et développement en vue de la réalisation de fonctions d'aide à l'orchestration : recherche des combinaisons d'échantillons (instruments) dont la superposition approche le plus une cible donnée ; développement de versions successives de prototypes applicatifs, dans l'environnement d'aide à la composition OpenMusic de l'Ircam, destinée aux compositeurs ; Coordination : Ircam. Participants : Ircam-AnaSyn, Ircam-RepMus.
- *SP5-Echantillonneur* : développement de versions successives l'application d'échantillonneur, intégrant les fonctions développées dans les autres sous-projets. Coordinateur : Univers Sons. Participants : Univers Sons, Ircam-AnaSyn, Ircam-IMTR.

Les sous-projets SP2 et SP3, ainsi que le début du sous-projet SP4, relèvent d'activités de rercherche et de développement visant la réalisation de fonctions spécifiques ; les sous-projets SP4 (phase finale) et SP5 concernent des activités de développement des deux applications cibles.

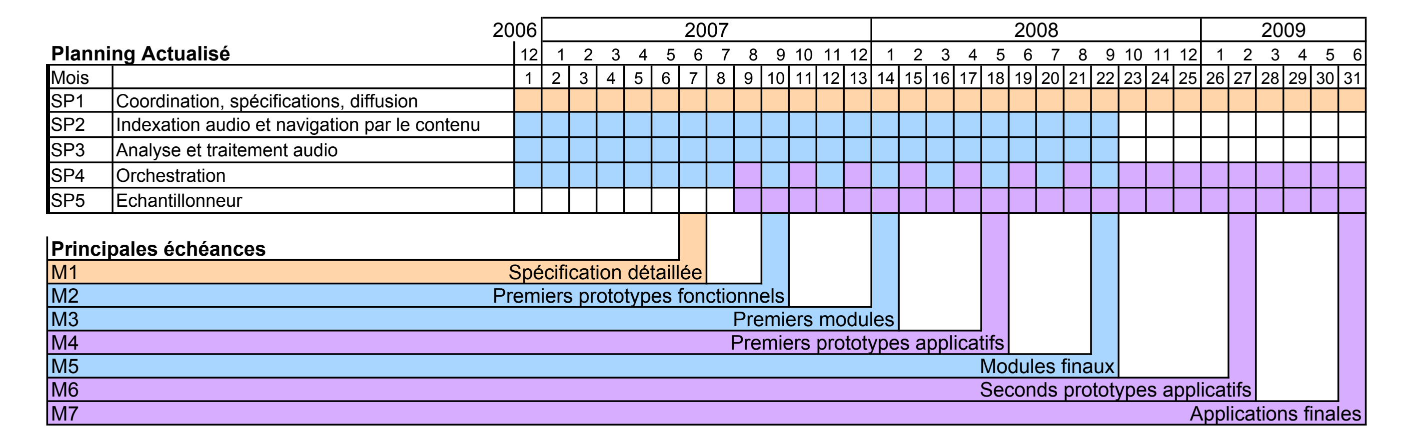
---

[1] http://www.ircam.fr

[2] http://www.univers-sons.com/



Le planning initial du projet se basait sur une période de 24 mois, mais dès la spécification détaillée menée au premier semestre du projet, une extension à 30 mois s'est avérée nécessaire selon le planning ci-après :



Ce planning a été respecté dans l'ensemble de ses échéances principales. Certaines tâches de recherche relatives au SP2 et au SP3 se sont poursuivies au dernier semestre du projet.

1.2 Spécification détaillée

Parallèlement au démarrage des tâches de recherche, l'ensemble des participants ont été mobilisés au cours des 6 premiers mois du projet en vue de la production d'une spécification détaillée (délivrable), comportant les étapes suivantes :

- Etat de l'art scientifique, recensement de l'ensemble des éléments de technologie issus de l'Ircam et propositions de fonctionnalités destinées à alimenter la spécification fonctionnelle ;
- Etude des marchés visés (applications d'échantillonneurs logiciels) et analyse des besoins pour les deux prototypes applicatifs (Aide à l'orchestration, Echantillonneur) ;
- Spécification fonctionnelle des deux prototypes applicatifs ;

- Spécification technique des deux prototypes applicatifs, notamment architecture logicielle et choix d'implémentation ;
- Elaboration d'un plan de travail détaillé (sous-tâches, délivrables internes, planning détaillé) ;

Ces éléments élaborés en commun ont servi de base à l'ensemble des travaux du projet.

1.3 Indexation audio et navigation par le contenu

1.3.1 Introduction

L'ensemble des travaux réalisés dans ce sous-projet ont visé la réalisation de fonctions de sélection et de navigation dans des bases de données d'échantillons sonores, à partir de l'analyse des signaux sonores (fichiers sons). Ces fonctions se décomposent selon les modules suivants (cf Figure 1) :

- Extraction de descripteurs audio (§ 1.3.2) : ensemble exhaustif de descripteurs de bas niveau caractérisant les différents aspects du contenu sonore des sons. Ces descripteurs sont utilisés pour le calcul des fonctions suivantes d'indexation de plus haut niveau et une partie d'entre eux est directement utilisable dans l'interface de navigation offerte à l'utilisateur.
- Classification automatique d'échantillons : à partir de jeux de classes prédéfinis, le système, par analyse du fichier son, affecte une classe à tout nouvel échantillon. Les travaux menés à cet effet ont comporté plusieurs étapes : constitution d'un corpus de sons représentatifs de sons dits d' « environnement », c'est-à-dire de bruitages, fournis par Univers sons et études en psychologie de l'audition visant à caractériser les classes pertinentes à partir de sons (§ 1.3.6, 1.3.7, 1.3.8). Une fois les classes définies, les travaux de recherche ont porté sur la mise en œuvre de fonctions de classification automatique à partir des descripteurs audio (§1.3.3). Les trois principaux jeux de classes pour lesquels des fonctions d'indexation automatique ont été développés ont concerné :
  - La description « causale » des sons, c'est-à-dire de l'origine de leur production ;
  - Une description sous forme d'onomatopées
  - La description morphologique des sons, sur la base des théories de Pierre Schaeffer ;
- Recherche par similarité : à partir d'un son de départ donné par l'utilisateur, rechercher les sons les plus proches selon une mesure de similarité sonore pertinente (§1.3.3.4)
- Caractérisation automatique de boucles rythmiques : types d'échantillons particuliers correspondant à une période de sons pulsés destinés à s'intégrer dans une séquence rythmique (§1.3.4).
- Mise au point d'un prototype d'application fournissant une interface graphique de navigation dans une ou plusieurs bases de données d'échantillons et produisant une synthèse sonore en temps réel en fonction de la séquence de navigation (§1.3.9).

La présentation des travaux effectués et leurs résultats est donnée selon la structure du programme de travail détaillé du projet défini dans son document de spécification détaillée, et non dans l'ordre logique de réalisation des tâches.

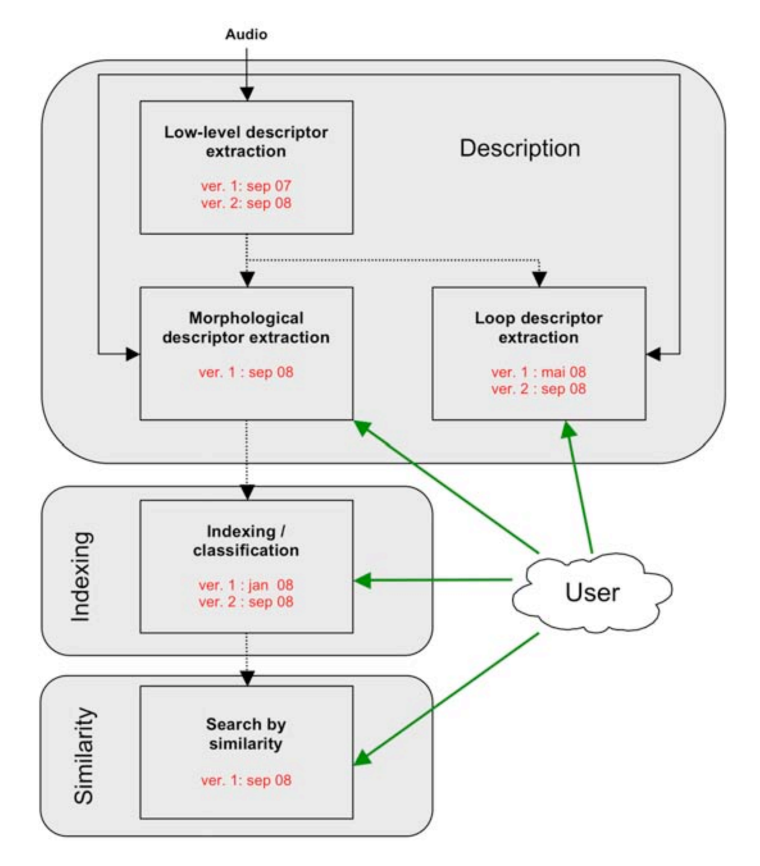


Figure 1 - Modules fonctionnels du sous-système d'indexation automatique des échantillons

1.3.2 SP2-1 – Module d'extraction de descripteurs audio pour les échantillons

Responsable : Ircam-AnaSyn

L'extraction des descripteurs de bas niveau est effectuée à partir de l'outil `ircamdescriptor` et de sa bibliothèque associée IrcamDescriptor. La version finale se présente sous forme d'une bibliothèque C++, qui a initialement été développée sous la forme d'un code Matlab, issu d'autres projets, qui implémente l'ensemble des 54 descripteurs (166 si l'on considère chaque dimension comme un descripteur distinct) décrits dans [Peeters04].

Les principaux travaux réalisés ont ainsi consisté en:

- la définition d'un ensemble complet et cohérent de descripteurs
- une écriture complète du code C++, selon une nouvelle architecture optimisée;
- une implémentation complète, comme format de sortie, du standard SDIF[3], co-développé par plusieurs laboratoires de recherche dans le monde spécialisés dans les technologies audio, et utilisé comme format de fichier pour stocker, notamment sous forme binaire, tous types d'analyses audio indexées sur le temps.

1.3.2.2 Définition d'un ensemble cohérent et complet de descripteurs

La liste complète des descripteurs calculés avec leur type SDIF est donnée à la Figure 41 p.80. Ils comprennent des descripteurs instantanés et des descripteurs "temporels" résultant d'un calcul sur un intervalle de temps donné de descripteurs instantanés.

L'implémentation initiale de ircamdescriptor et de son équivalent C++ était conçue au départ pour une tâche de classification d'échantillons instrumentaux jouant des notes isolées : en général, ce type de signaux ont un comportement bien maîtrisé : variations lentes et bornées

[3] http://sdif.sourceforge.net/

d'énergie, spectres quasi-harmoniques. L'objectif était ici plus délicat, car il portait sur des sons de bruitage, sur lesquels peu d'hypothèses pouvaient être faites a priori. Une part conséquente du travail réalisé a ainsi consisté à rendre les caractéristiques calculées suffisamment robustes pour convenir à de larges classes de sons.

1.3.2.3 Implantation de SDIF comme format de sortie

Un grand nombre de nouveaux types SDIF a été déclaré pour permettre l'extension du format nécessaire au stockage de tous les descripteurs calculés, et un cadre a été défini pour la modélisation temporelle des descripteurs instantanés. Ces travaux ont été présentés pour discussion à la communauté Music Information Retrieval [Burred08].

Dans le cas général, SDIF stocke des matrices 2D indexées sur le temps, et dans le cas des descripteurs, les lignes ont été utilisées pour les coordonnées de descripteurs et les colonnes pour les différentes variantes de calcul du descripteur.

La Figure 2 donne les différentes variantes possibles des descripteurs.

	variations
3-variation descriptors	1. Linear amplitude 2. Power amplitude 3. Log. amplitude
6-variation descriptors	1. Linear amplitude, linear frequency 2. Power amplitude, linear frequency 3. Log. amplitude, linear frequency 4. Linear amplitude, log. frequency 5. Power amplitude, log. frequency 6. Log. amplitude, log. frequency

Figure 2 - Signification des variantes des descripteurs

Un modèle a été défini pour permettre différentes représentations temporelles de descripteurs sur un intervalle de temps donné, s'ajoutant à leur représentation instantanée. Les modèles temporels existants, avec leur type SDIF correspondant, sont donnés à la Figure 3. Le format permet ainsi de superposer différentes représentations temporelles d'un même descripteur dans un même fichier SDIF.

	SDIF type	**applicable to**
LoudnessWeightedMean	1WMN	all
LoudnessWeightedStandardDeviation	1WSD	all
ModulationAmplitude	1MDA	1FQ0,1EEV
ModulationFrequency	1MDF	1FQ0,1EEV
LogAttackTime	1LAT	1EEV,1LDN
TemporalIncrease	1TIN	1EEV,1LDN
TemporalDecrease	1TDE	1EEV,1LDN
TemporalCentroid	1TCN	1EEV,1LDN
EffectiveDuration	1EFD	1EEV,1LDN

Figure 3 Modèles temporels existants avec leurs signatures SDIF

Une description plus détaillée des extensions opérées au format SDIF pour les descripteurs audio est accessible à : http://sdif.sourceforge.net/descriptor-types .

1.3.2.4 Implantation et nouvelle API de la bibliothèque d'extraction

Partant du code Matlab existant, une nouvelle bibliothèque, appelée IrcamDescriptor (dernière version développée : V2) a été développée de zéro, avec une nouvelle architecture conçue selon des critères d'optimisation pour une utilisation temps réel, de sécurité, d'extensibilité, de flexibilité et de facilité de maintenance.

Les besoins de rapidité, de flexibilité et de modularité ont constitué un défi important. La nouvelle bibliothèque a été conçue à cet effet selon des techniques C++ avancées comme la métaprogrammation (type-lists, type-info wrappers, etc.), conception policy-based, vérification de dépendances à la compilation, etc. Le point-clé de la bibliothèque est de fournir un dispositif configurable dans lequel chaque descripteur connaît ses propres dépendances, et la bibliothèque est à la fin capable de construire automatiquement un graphe de calcul reposant sur ces dépendances. En effet, le calcul de descripteurs passe par l'appel de fonctions communes, telles que des FFT (transformées de Fourier Rapide) fenêtrées sur un intervalle de temps donné, et l'explicitation et la gestion de ces dépendances permet à la fois l'optimisation et la modularisation des calculs.

Le calcul de chaque descripteurs repose sur deux policies qui définissent son comportement : une *policy* de séquencement et une *policy* de calcul (appelée contexte polyadique). A partir du moment où ces *policies* sont fournies, il est très facile de les combiner de nouvelles manières pour créer un nouveau descripteur. La Figure 4 fournit une description approximative de ce design.

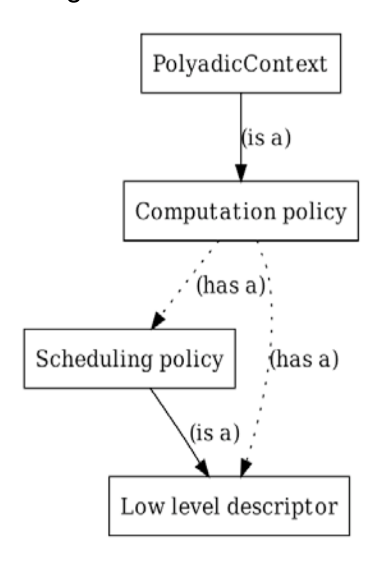


Figure 4 - Policy-based design

1.3.3 SP2-2 – Module d'indexation automatique

Responsable : : Ircam-AnaSyn

1.3.3.2 Classification automatique

Ce module de classification automatique prend en entrée un fichier son et produit en sortie un nom de classe et un ensemble de probabilités pour chaque classe. L'effort dans le cadre du projet s'est concentré sur les sons environnementaux (bruitages). La classification est supervisée : la taxonomie de classes est prédéfinie et la conception du système repose sur une phase d'apprentissage utilisant une base de sons annotée manuellement. Il a été décidé au cours de l'exécution du projet de ne pas rendre accessible l'apprentissage à l'utilisateur final du logiciel de classification (ingénieur du son, musicien). L'apprentissage est ainsi effectué entièrement en utilisant l'outil Matlab `ircamclassifier` - à ne pas confondre avec l'outil C++ de même nom et sa bibliothèque associée, qui ne fait que la classification- qui stocke le modèle soit sous la forme d'un fichier SDIF spécifique, soit d'un fichier texte. Ces fichiers sont alors lus par l'outil C++, avec les valeurs de descripteurs du fichier son, pour l'assignation d'un label et le calcul des probabilités.

Vue d'ensemble du processus d'apprentissage

Le module d'apprentissage en Matlab se base sur le classificateur issu du projet CUIDADO, comme décrit dans [Peeters03]. Les sous-modules de l'apprentissage sont les suivants :

1. **Prétraitement**. Comporte les étapes suivantes :
    1. **Normalisation des descripteurs** basée sur l'intervalle Inter-quartile (*Inter-Quartile Range - IQR*). Cette méthode de normalisation est plus robuste vis-à-vis des données extrêmes (*outliers*) que la normalisation basée sur la variance.
    2. **Transformation Box-Cox** (optionnelle) pour améliorer la gaussianité des classes.
    3. **Sélection manuelle des descripteurs**. Pour le moment, Fluctuation Strength, Roughness and Noise Shape Features, qui sont les seuls descripteurs qui ne sont pas implémentés dans la version C++, sont écartés.
    4. **Suppression des outliers** (optionnel).
    5. **Réduction de dimensionnalité**. Nécessaire à la fois pour éviter la "malédiction de la dimension" (*curse of dimensionality*) et permettre l'adaptabilité du système à différentes tâches de classification. Comprend les étapes suivantes :
    6. **Sélection automatique des descripteurs**. Repose sur la méthode IRMFSP (*Inertia Ratio Maximization using Feature Space Projection* [Peeters03]).
    7. **Analyse en composantes principales (ACP)** (optionnel). Effectue une rotation de l'espace des descripteurs pour aligner les directions de variance maximale avec les axes. Réduit la dimensionnalité dans le sens d'une description optimale.
    8. **Analyse discriminante linéaire (ADL)** (optionnel). Effectue une rotation de l'espace des descripteurs pour maximiser la séparation entre les classes. Fournit au plus C-1 descripteurs, où C est le nombre de classes d'apprentissage.
2. **Apprentissage supervisé**. Le module d'apprentissage implémente les méthodes suivantes :
    1. **Modèle gaussien**. Modélisation gaussienne multivariée unimodale.
    2. **Modèle de mélange de gaussiennes (GMM)**. Se base sur l'algorithme d'apprentissage E-M (Expectation-Maximization) initialisé par une segmentation par k-Means.
    3. **Arbre binaire de réduction entropique (BERT)**.
    4. **K-plus proches voisins (KNN)**.
    5. **Support Vector Machines (SVM).**

Le module de classification comprend un environnement complet de validation croisée et appelle un module d'évaluation calculant les mesures de précision de classification (classification accuracy, recall, precision et F-Measure), ainsi que la matrice de confusion.

Premières expériences d'évaluation

Dans les phases initiales du projet, deux cycles d'expériences ont été menées. D'abord de manière préliminaire pour valider les principes de conception de base, puis un ensemble d'expériences plus détaillées et formalisées, basées sur une validation croisée, qui ont fourni une vue d'ensemble plus réaliste de la performance. Nous nous concentrerons ici sur les expériences finales d'évaluation, tout en rappelant quelques conclusions importantes issues de ces expériences initiales :

- la confirmation de faisabilité d'une classification par matériaux autour de 6 classes, et la supériorité des méthodes gaussienne et GMM sur kNN et BERT, qui ont été depuis écartées (SVM n'était pas encore implémenté);

- la supériorité de GMM sur les modèles gaussiens unimodaux, la faisabilité d'une classification statistique reposant sur des labels d'onomatopées, et l'importance de l'évaluation de la complexité du modèle (nombre de clusters gaussiens, type de la matrice de covariance). Les meilleurs précisions de classification obtenues en utilisant la méthode GMM étaient de 72.38% pour la base de données à 6 classes de matériaux et de 70.74% pour la base de 5 classes d'onomatopées.

Redéfinition des taxonomies

Les taxonomies de matériaux et onomatopées citées plus haut étaient définies comme un sous-ensemble de la taxonomie de 46 classes d'exemples fournie par Univers sons, qui n'était pas adaptée à un apprentissage statistique du fait du grand nombre de classes, des populations non équilibrées d'échantillons prototypes et de critères ambigus. Sur la base des travaux de l'équipe PDS (cf § 1.3.7 p.31), les taxonomies ont été redéfinies. Le détail des critères de cette redéfinition est donné dans le délivrable SP2-M22.

Les taxonomies finales sont listées en Figure 5. Tous les échantillons utilisés sont issus des bases de données commerciales de bruitages Hollywood Edge, Blue Box, SoundIdeas and Soundscan.

Taxonomie	**Causale**		**Causale réduite**		**Onomatopées**	
Nbre de classes	7		5		5	
Précision de choix aléatoire	14,29%		20,00%		20,00%	
Nom des classes et nombre de fichiers	1. Electrical	66	1. Electrical	66	1. Buzzing	124
	2. Fire	64	2. Gas	135	2. Crackling	159
	3. Gas	135	3. Liquid	269	3. Rushes	154
	4. Liquid	269	4. Solid-metal	192	4. Scratching	48
	5. Solid-metal	192	5. Solid-paper	67	5. Squeaking	106
	6. Solid-paper	67				
	7. Solid-wood	103				
Nbre total de fichiers son		896		729		591

Figure 5 – Taxonomies finales

Expériences d'évaluation finales

L'objectif de ces expériences était de trouver la méthode de classification et ses paramètres optimaux pour les trois modèles de classification : deux pour la taxonomie causale (complète et réduite) et un pour la taxonomie d'onomatopées. Le détail de la méthode est décrit dans le délivrable SP2-M22. Pour toutes les méthodes testées, les descripteurs ont été calculés à partir de l'implémentation C++ de ircamdescriptor. Pour la méthode SVM, l'implémentaiton `libsvm`4 a été choisie en raison de sa flexibilité, son efficacité (elle est implémentée en C++ et accessible par Matlab comme ensemble de fichiers `mex`), sa bonne documentation and sa politique de licence adaptée. Parmi les différentes définitions, la soft-margin SVM, aussi appelé C-SVM, a été utilisée. Ce modèle utilise un paramètre de coût *C* pour les pénalités de mauvaise classification et améliore la généralité. Le noyau choisi est RBF (*Radial Basis Function*), qui a l'avantage de n'être contrôlé que par un paramètre γ.

Les paramètres de modèles à optimiser étaient les suivants:

- Pour les modèles **Gaussiens** unimodaux: Nombre de descripteurs (de 1 à 40) présence/absence de ALD. L'optimisation était exhaustive.
- Pour les **GMM**: Nombre de descripteurs (de 1 à 40), nombre de clusters gaussiens par GMM (de 1 à 10), type de matrice de covariance (par ordre croissant de

4 `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

complexité: sphérique, diagonale, pleine) et présence/absence de ALD. L'optimisation était exhaustive.

- POUR **C-SVM-RBF**: Nombre de descripteurs (de 1 à 40), paramètre de coût *C* (de 2^{-5} à 2^{20}, en pas logarithmiques de 2^2) et paramètre de noyau RBF γ (de: 2^{-15}à 2^3, en pas logarithmiques de 2^3). Pour éviter une explosion combinatoire sur un ensemble de recherche 3D, la recherche a été faite en deux étapes : d'abord recherche du nombre optimal de descripteurs, puis optimisation 2D pour les couples (C, γ).

Au total, 9 configurations expérimentales différentes ont été prises en compte, comme indiqué en Figure 6. Ces résultats correspondent à un total de $(2 \times 40) + (6 \times 40 \times 10) + 40 + (13 \times 7) = 2611$ calculs de validation croisée pour chaque base de données, $2611 \times 5 = 13055$ calculs individuels de classification par base de données, soit un grand total de 13055*3 = 39165 calculs.

Quelques conclusions générales peuvent être tirées de ces résultats:

- C'est la méthode C-SVM qui obtient les meilleurs résultats pour les 3 bases de données, à la fois selon les mesures accuracy and F-measure. Les meilleures accuracies obtenues sont de 74,33% pour la taxonomie causale complète, 83,72% pour la taxonomie causale réduite et 72,21% pour la taxonomie d'onomatopées.
- GMM a obtenu de meilleures résultats que les gaussiennes unimodales. En moyenne, les matrices de covariance diagonales ont obtenu les meilleurs résultats, offrant un compromis adapté en terme de complexité.
- L'ALD réduit beaucoup la présence de singularités et la "malédiction de dimensionnalité" (le nombre optimal de descripteurs est plus élevé avec l'ALD).

			Causal (7 classes)				
Classif.	LDA	Options	**Max. ACC**	best par. (ACC)	**Max. FMSR**	best par. (FMSR)	% singu-larities
Gauss			67,78 ± 1,90	19	66,53 ± 7,64	20	0
Gauss	Y		66,99 ± 3,36	40	65,31 ± 6,79	40	0
GMM		Full cov.	68,81 ± 3,83	21/1	67,31 ± 7,58	20/1	11,25
GMM	Y	Full cov.	67,34 ± 3,80	40/2	65,05 ± 7,66	37/2	1,50
GMM		Diag. cov.	72,60 ± 6,33	30/7	70,55 ± 8,65	30/7	0,20
GMM	Y	Diag. cov.	68,01 ± 3,83	33/4	65,74 ± 6,47	40/3	0,20
GMM		Spher. cov.	69,71 ± 0,00	30/8	65,90 ± 0,00	30/8	30,40
GMM	Y	Spher. cov.	68,12 ± 2,20	35/5	65,80 ± 8,51	40/3	0
C-SVM		RBF kernel	**74,33** ± 2,31	30/3/-6	**72,35** ± 5,22	30/11/-6	0
			Causal-réduit (5 classes)				
Classif.	LDA	Options	**Max. ACC**	best par. (ACC)	**Max. FMSR**	best par. (FMSR)	% singu-larities
Gauss			75,14 ± 2,94	16	74,49 ± 6,99	24	0
Gauss	Y		76,81 ± 4,50	37	75,50 ± 6,31	37	0
GMM		Full cov.	78,03 ± 5,53	17/2	76,21 ± 0,00	32/10	22,70
GMM	Y	Full cov.	77,80 ± 3,60	32/2	76,95 ± 5,78	32/2	0,75
GMM		Diag. cov.	81,77 ± 3,87	25/6	79,18 ± 9,13	25/6	0,25
GMM	Y	Diag. cov.	77,67 ± 4,97	32/2	76,56 ± 6,20	35/5	0,05
GMM		Spher. cov.	79,31 ± 0,00	19/9	76,92 ± 0,00	12/9	31,10
GMM	Y	Spher. cov.	76,24 ± 2,47	35/8	74,55 ± 5,27	39/9	0
C-SVM		RBF kernel	**83,72** ± 4,89	20/17/-3	**80,78** ± 7,56	20/17/-3	0

			Onomat (5 classes)					
Classif.	LDA	Options	**Max. ACC**	best par. (ACC)	**Max. FMSR**	best par. (FMSR)	% singu-larities
Gauss			68,34 ± 3,89	26	62,02 ± 6,79	19	0
Gauss	Y		68,45 ± 3,40	29	64,69 ± 6,30	26	0
GMM		Full cov.	71,43 ± 1,19	37/2	65,05 ± 3,43	37/2	6,40
GMM	Y	Full cov.	69,01 ± 2,72	37/1	65,29 ± 7,96	40/4	1,10
GMM		Diag. cov.	68,80 ± 3,39	22/9	64,55 ± 10,12	27/8	0,40
GMM	Y	Diag. cov.	68,47 ± 2,53	35/3	65,20 ± 8,82	36/7	0
GMM		Spher. cov.	65,79 ± 0,00	39/5	62,14 ± 0,00	39/5	31,15
GMM	Y	Spher. cov.	69,82 ± 3,32	40/9	65,54 ± 6,25	40/9	0
C-SVM		RBF kernel	**72,21** ± 3,74	20/7/-3	**69,24** ± 7,22	20/7/-3	0

Figure 6 – Ensemble de résultats des expériences d'apprentissage : maximum average accuracy ACC en % and F-measure F, ± variance

Les Figure 43, Figure 44 et Figure 45 présentées p. 81 dans la partie relative aux résultats scientifiques donnent les matrices de confusion obtenues pour les paramètres optimaux.

Les modèles de SVM finalement livrés ont été entraînés en utilisant la base de données complète (sans partitionnement entre entraînement et test) avec les paramètres optimaux obtenus pendant les test. Les précisions obtenues de cette manière ont été proches de 90% pour les 3 taxonomies. Il doit cependant être noté que ce chiffre ne peut être pris comme mesure de performance car il viole la règle de validation croisée.

1.3.3.3 Description et classification morphologiques

Deux critères ont été envisagés au cours du projet pour la description morphologique des sons : les profils dynamiques et les profils mélodiques (de hauteur). Les tentatives pour définir des classes de profils mélodiques stables et convaincants se sont avérées infructueuses [Peeters08] et il a dont été décidé de concentrer les efforts sur les profils dynamiques pour la classification automatique, reposant sur les classes suivantes conjointement élaborées par les équipes Analyse/synthèse et PDS (cf §1.3.6.2) : *stable*, *train d'impulsions*, *décroissant*, *impulsif*, *croissant/décroissant* et *croissant*.

De telles classes dynamiques peuvent être décrites en mesurant un certain nombre de pentes et durées à partir d'une approximation par morceaux d'une enveloppe temporelle. Des expériences préliminaires décrites dans [Peeters08] se basaient sur des approximation par splines. Les possibilités de modélisation temporelle de IrcamDescriptor ont permis d'appliquer des approximiations similaires aux segments AR (attack/rest) d'une des enveloppes temporelles disponibles comme descripteurs à court terme : l'énergie totale (`1NRG`), l'enveloppe en énergie (`1EEV`) and la sonie (`1LDN`). A cet effet, les modélisations suivantes sont disponibles :

- **Temporal increase** (`1TIN`): moyenne pondérée de segments de pente d'enveloppe pendant la phase d'attaque. C'est une approximation linéaire de l'enveloppe d'attaque.
- **Temporal decrease** (`1TDE`): exposant d'une approximation exponentielle décroissante de l'enveloppe pendant la phase de maintien, estimée par une régression linéaire du logarithme de l'une des enveloppes temporelles possibles.
- **Effective duration** (`1EFD`): temps pendant lequel l'enveloppe est plus supérieure à un seuil relatif de 40%.

La Figure 7 montre un exemple d'approximation des segments AR de l'enveloppe et la durée effective d'un son montant/descendant, dans ce cas basé sur l'enveloppe d'énergie totale.

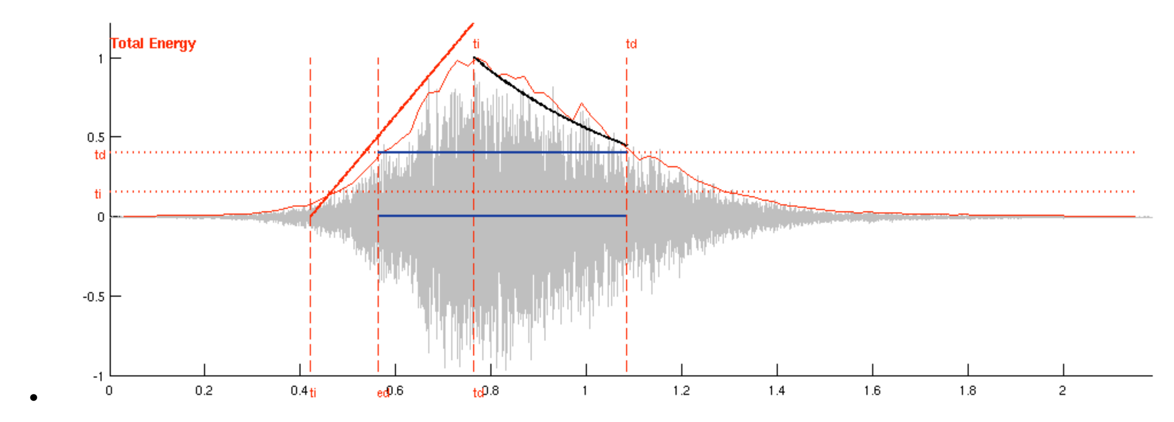


Figure 7 – Approximation par morceaux de l'enveloppe temporelle pour la description morphologique de profils dynamiques

Modèle de classification pour les profils dynamiques

Le problème de classification morphologique est plus simple que les tâches de classification statistique (causale, onomatopées) présentées dans la section précédente. Comme seulement trois critères sont présents, et qu'ils ont un lien sémantique direct avec les définitions des classes, il n'y a pas besoin de transformations à grande échelle de descripteurs, de sélection de descripteurs et d'estimation de densités ou de paramètres de modèle. La classification peut être directement implantée sous la forme d'un ensemble de règles liées à des seuils fixes de valeurs de descripteurs. A cet effet, nous avons utilisé un algorithme par arbre utilisant l'index de diversité de Gini comme critère de décision (cet index mesure l'équilibre d'impuretés pur chaque sous-branche candidate à séparer ; dans ce cas, l'impureté correspond au nombre de mauvaises classifications, supposées de coût égal).

Pour l'apprentissage et l'évaluation, une base de données annotée manuellement de 187 sons représentatifs a été utilisée. La Figure 8 montre les résultats de précision de classification. Le meilleur (93,05%) a été obtenu avec l'enveloppe de sonie.

Envelope type	Précision de classification (5 classes, hasard : 20%)
Sonie	**93,05 %**
Energie totale	90,91 %
Enveloppe d'énergie	92,51 %

Figure 8 – Précision de classification pour la description morphologique de profils dynamiques

La Figure 9 montre l'arbre optimal obtenu pour la sonie.

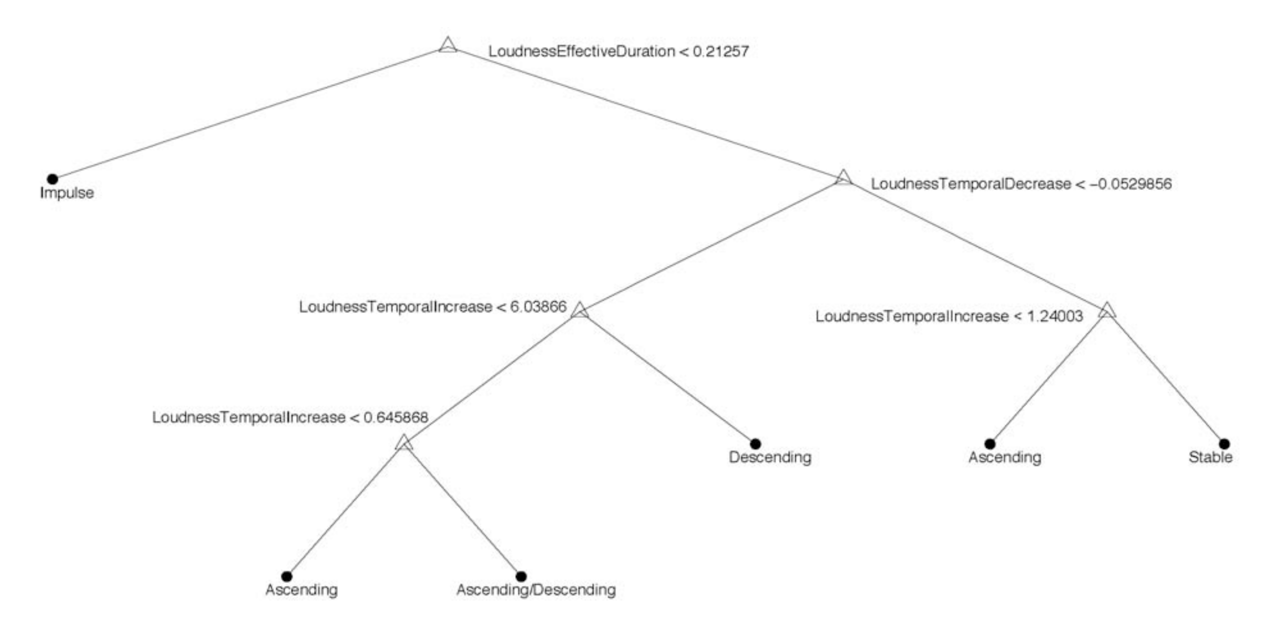


Figure 9 - Arbre optimal de classification pour la description morphologique des profils dynamiques

1.3.3.4 Recherche par similarité

Un certain nombre de tests informels ont été effectués pour démontrer la possibilité d'indexer une base de données pour la recherche par similarité à l'aide de l'outil de classification IrcamClassifier. Celui-ci permet de produire les mesures de similarité selon méthodes différentes, reposant toutes deux sur la connaissance des classes statistiques considérées (causale, causale réduite, onomatopées) :

- calcul des similarités à partir des probabilités des classes obtenues après classification. L'inconvénient est que la définition de similarité dépend d'un modèle de classification particulier ;
- similarité sur un ensemble réduit de descripteurs : les coordonnées de similarité sont obtenues par une transformation de l'espace des descripteurs vers un espace de dimensionnalité réduite par ALD, par exemple en deux dimensions. Pour les taxonomies causal réduit et onomat, cela donne les plans de similarité présentés en Figure 10 et Figure 11. Cela est notamment utilisé dans l'application CataRT.

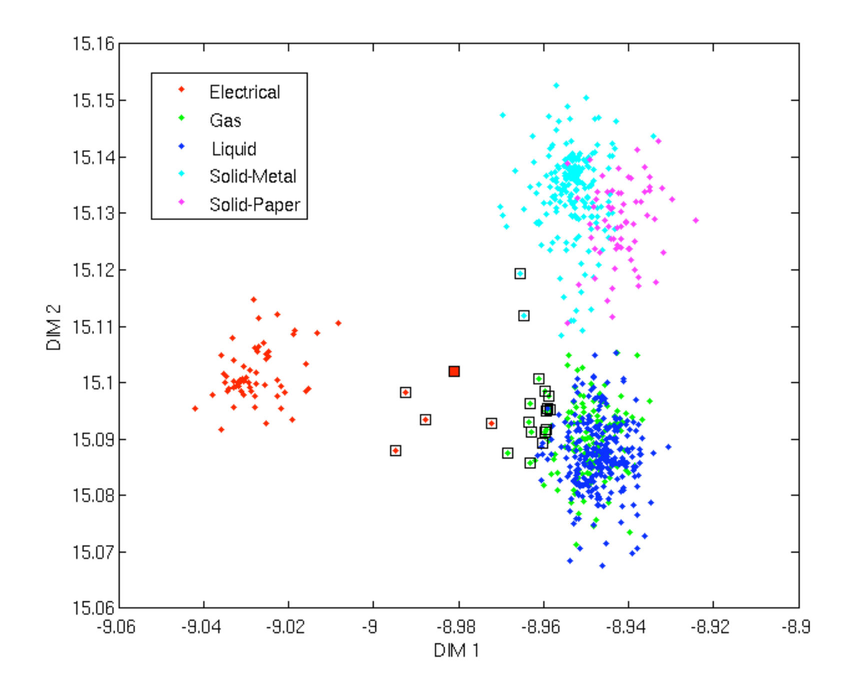


Figure 10 - Plan de similarité par ALD pour la base causale réduite. Le carré plein vers le centre est un exemple de requête, et les points avec des tours carrés correspondent aux 20 plus proches voisins (distance euclidienne)

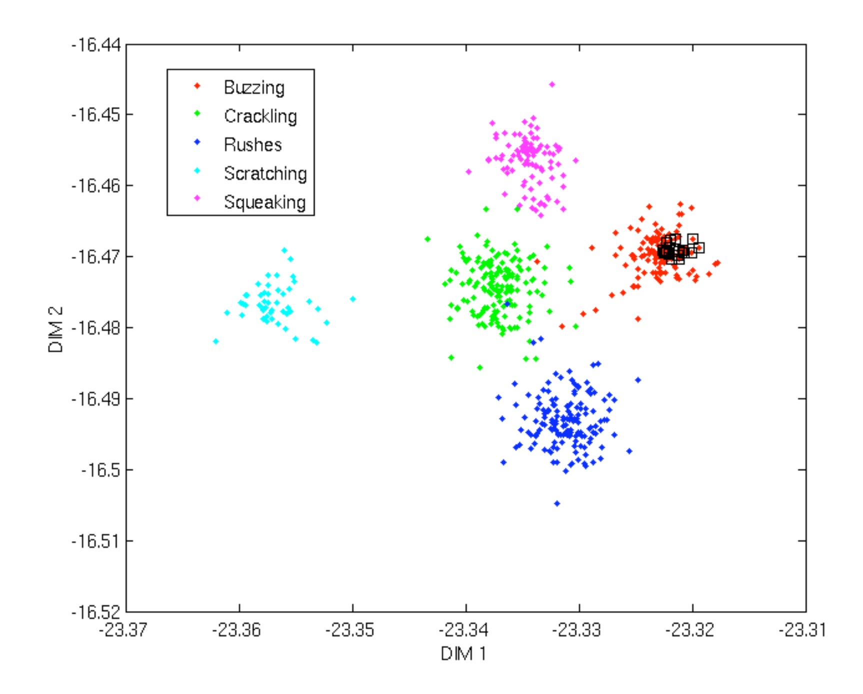


Figure 11 - Plan de similarité par ALD pour la base causale réduite; même visualisation des résultats de recherche que pour la figure précédente.

1.3.4 SP2-3 – Module d'extraction de descripteurs audio pour les boucles

Responsable : : Ircam-AnaSyn

Les boucles sont décrites selon deux points de vue : le contenu rythmique et harmonique.

Le contenu rythmique est décrit par ircambeat, qui fournit le tempo moyen (bpm), la métrique moyenne (either 22 pour les mesures 2/4 et 4/4, 23 pour les mesures 3/4 ou 32 pour les

mesures 6/8), un indice de périodicité, un indice de percussivité, et un pattern rythmique à 18 dimensions décrivant l'énergie présente aux différentes subdivisions et multiples de la pulsation[PeetersISMIR2005]. L'algorithme fournit ses résultats sous forme de fichier XML, dont un exemple est donné ci-après:

<?xml version="1.0" ?>
<rhythmdescription>
<media>***</media>
<description>
<bpm>141.0437920374</bpm>
<meter>22.0000000000</meter>
<percussivity>2.7931836821</percussivity>
<periodicity>0.9905488875</periodicity>
<rhythmpattern>… </rhythmpattern>
</description>
</rhythmdescription>

L'algorithme Ircambeat est décrit dans [PeetersEurasip2007] et repose sur l'algorithme donné en Figure 12.

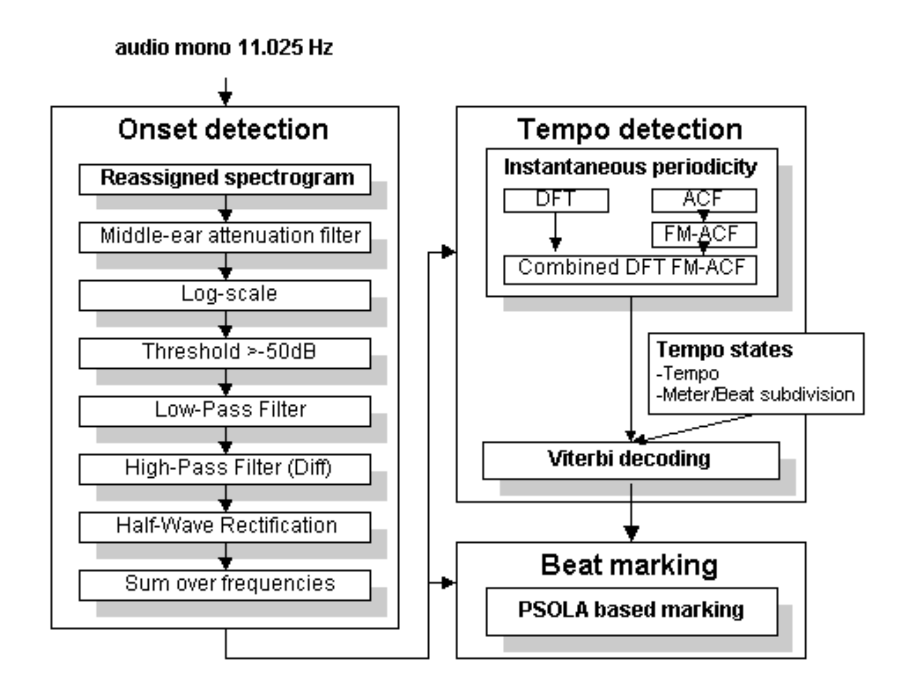


Figure 12 - Algorithme de calcul de ircambeat

La description harmonique est fournie par ircamkeymode, qui donne la tonalité moyenne (C, Db, D, Eb, …), le mode moyen (majeur, mineur), et un pattern harmonique qui fournit les probabilités de chaque tonalité/mode. Les résultats sont fournis sous forme de fichier XML, dont un exemple est donné ci-après:

<?xml version="1.0" ?>
<harmonicdescription>
<media>***</media>
<description>
<key>Eb</key>

```
    <mode>min</mode>
    <harmonicpattern>***</harmonicpattern>
  </description>
</harmonicdescription>
```

L'algorithme utilisé pour Ircamkeymode est décrit dans [PeetersISMIR2006] et présenté en Figure 13.

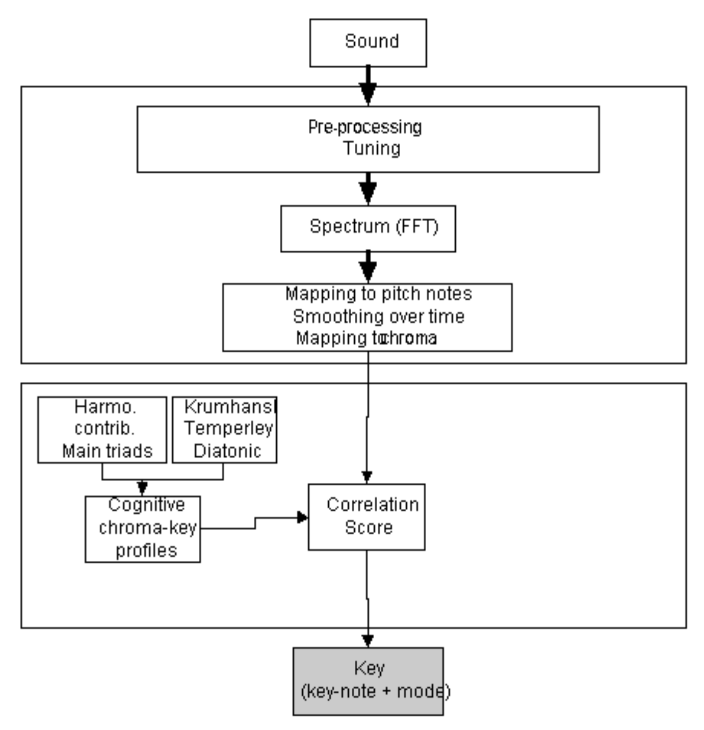


Figure 13 - Algorithme de calcul de ircamkeymode

<u>Travaux de développement effectués</u>

Les bibliothèques C++ *IrcamBeat* et *IrcamKeyMode*, issues du projet européen SemanticHIFI (2004-2006), ont été livrées pour MacOSX et Windows. Plusieurs travaux ont été nécessaires pour permettre leur fonctionnement :

- mise à jour à partir de la dernière version d'une bibliothèque MatMTL qu'elles utilisent et de ses extensions ;
- réalisation de modules d'interface (décrits ci-après) ;
- groupement de code commun dans une nouvelle bibliothèque *IrcamCommon*, qui expose l'interface à l'API utilisateur *AbstractReader.*

Interface généralisée

Afin de réduire le nombre d'opération et de simplifier l'API utilisateur, une interface commune a été développée. La Figure 14 montre le schéma de calcul développé, et valable pour toutes les bibliothèques livrées dans le cadre du SP2. Pour chaque bibliothèque, un *Computation Module* a été livré. L'utilisateur n'a qu'a le créer, le configurer et appeler la fonction *compute().* A la fin du calcul, l'utilisateur peut optionnellement sauvegarder les données calculées dans un fichier dans un format dépendant de la bibliothèque. La fonction *compute()* prend en entrée un pointeur sur un objet *AbstractReader* et rend un pointer sur une structure dépendante de la bibliothèque.

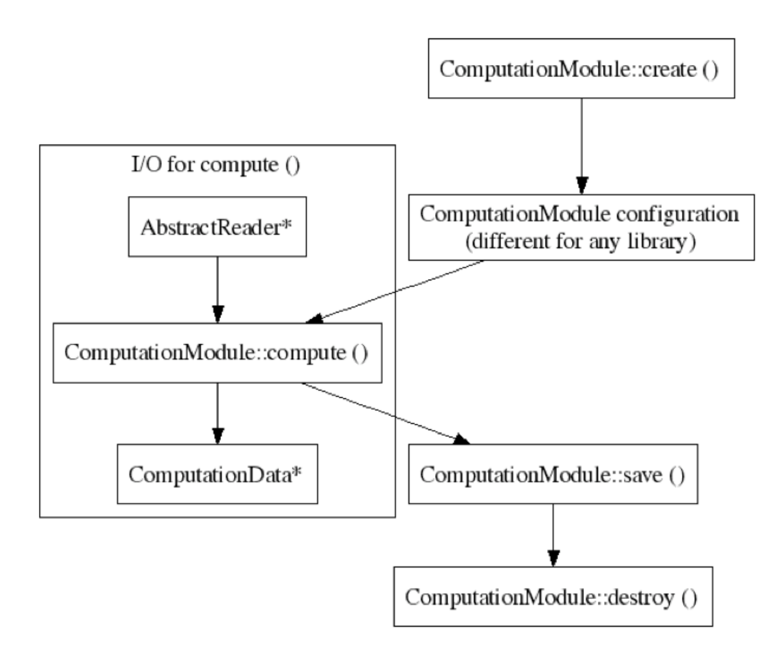


Figure 14 - Interface de programmation généralisée

1.3.5 Références SP2-1, SP2-2 et SP2-3

[Burred08] J.J. Burred, C.E. Cella, G. Peeters, A. Röbel and D. Schwarz, « Using the SDIF Sound Description Interchange Format for Audio Features», Int. Conf. On Music Information Retrieval (ISMIR), Philadelphia, USA, September 2008.

[Burred09] J.J. Burred, A. Röbel, T. Sikora, « Polyphonic Musical Instrument Recognition Based on a Dynamic Model of the Spectral Envelope », IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Taipei, Taiwan, 2009

[Peeters03] G. Peeters, « Automatic classification of large musical instrument databases using hierarchical classifiers with inertia ratio maximization », Proc. 115th Convention of the Audio Engineering Society (AES), New York, USA, October 2003.

[Peeters04] G. Peeters, « A large set of audio features for sound description (similarity and classification) in the CUIDADO project », Technical Report, IRCAM, 2004.

[Peeters08] G. Peeters and E. Deruty, « Automatic morphological description of sounds », Proc. Acoustics 08, Paris, France, June 2008.

[PeetersEurasip2007] Peeters, G. (2007). “Template-based estimation of time-varying tempo.” EURASIP Journal on Advances in Signal Processing 2007(Special Issue on Music Information Retrieval Based on Signal Processing): Article ID 67215, 14 pages.

[PeetersISMIR2005] Peeters, G. (2005). Rhythm Classification using spectral rhythm patterns. Proc. of ISMIR, London, UK.

[PeetersISMIR2006] Peeters, G. (2006). Chroma-based estimation of musical key from audio-signal analysis. Proc. of ISMIR, Victoria, Canada.

[SORdeliverableM10] Projet Sample Orchestrator, Délivrable M10, Premiers prototypes fonctionnels, Octobre 2007.

[SORdeliverableM18] Projet Sample Orchestrator, Délivrable M18, « Rapport interne d'avancement SP2», Juin 2008.

[SORdeliverableM22] Projet Sample Orchestrator, Délivrable M22, « Modules d’indexationfinaux», Octobre 2008.

[SORminutes1] Sample Orchestrator Project, Minutes of the 1st SP2 Meeting, Ircam, 23rd Août 2007.

1.3.6 SP2-4 – Description des sons environnementaux

1.3.6.1 D-SP2-4-1 – Validation de descripteurs audio pour les sons environnementaux

Responsable : Ircam-PDS

Objectifs

Le cadre de cette tâche est d'amorcer la réflexion sur une transposition/extension de la notion de descripteurs psychoacoustiques, telle qu'elle est développée pour les sons musicaux, à un ensemble de sons environnementaux qualifiés, par nature, de non-musicaux. L'objectif est donc, d'une part, de généraliser la notion d'espace de timbre à des modèles hybrides (catégoriels / continus), et d'autre part, de proposer des outils de prédiction – notamment, de classification automatique – adaptés à ces nouvelles représentations afin d'apporter des réponses aux nécessités de description et de taxonomie de ce type de sons.

Etat de l'art

Ce travail s'appuie en premier lieu sur les travaux de référence dans le domaine du timbre musical [Grey77], [Krumhansl89], [Krimphoff94a], [McAdams95a], et plus spécifiquement sur quatre études du timbre de sons du quotidien : **A-** habitacles de voiture [McAdams98c], **B-** unités de climatisation [Susini04b], **C-** klaxons [Lemaitre07a] et **D-** portières automobile [Parizet06].

Ces quatre études ont été notamment choisies pour la méthodologie commune qu'elles déroulent et qui autorise la comparaison des résultats ; la démarche comporte les éléments principaux suivants :

- Génération du corpus (incluant notamment, pré-classification et égalisation en sonie) ;
- Expérience de dissimilarité (jugement de la dissemblance entre deux sons) ;
- Analyse multidimensionnelle MDS (CLASCAL [Winsberg93] ou INDSCAL [Caroll70]) aboutissant aux axes perceptifs de l'espace de timbre) ;
- Corrélation de ces axes avec des descripteurs acoustiques et psychoacoustiques.

Les données propres à chaque étude – qui servent de données d'entrée au travail en question – peuvent être synthétisées dans un tableau général (cf.Figure 15– se reporter respectivement à [McAdams98c], [Susini04b], [Lemaitre07a] et [Parizet06] pour la définition et la formulation des descripteurs correspondants à chacune des études).

	A- Habitacles voiture	**B- Unités climatisation**	**C- Klaxons**	**D- Portières auto.**
Corpus	A1 : 16 sons, 3eme, 4000 tr/min A2 : 14 sons, 5eme, 3500 tr/min	B : 19 sons de convecteurs (4 synthèses)	C : 22 sons avertisseurs	D : 12 sons portière
Analyse	CLASCAL	CLASCAL	CLASCAL	INDSCAL
Résultats	A1 : 3 dim., spécif., 1 class. A2 : 2 dim., spécif., 1 class.	B : 3 dim., spécif., 5 class.	C : 3 dim., spécif., 6 class.	D : 3 dim.
Descripteurs	A1, dim.1 : RAPh/nh –*A* A1, dim.2 : CGSERB A1, dim.3 : dec-BF A2, dim.1 : RAPh/nh –*A* A2, dim.2 : CGS –*C*	B, dim.1 : RAPh/nh –*A* B, dim.2 : CGSnh –*B* B, dim.3 : Sonie (Zwicker)	C, dim.1 : rugosité C, dim.2 : CGS -Gamm. C, dim.3 : dev	D, dim.1 : CGS D, dim.2 : « netteté » D, dim.3 : …

Figure 15 - Méta-analyse du timbre – tableau récapitulatif

Problématique

A partir de ces résultats, le travail consiste en une **méta-analyse** du timbre d'un sous-ensemble de sons environnementaux représenté par l'union des corpus de chacune des

études de référence. L'hypothèse de départ est celle d'une organisation à deux niveaux (cf. Figure 16 pour illustration) :

- un niveau catégoriel (discret), représenté par des grandes classes d'évènements sonores correspondant chacune à un type de source et une cause physique distincts.

- un niveau continu qui associe à chacune des classes un espace perceptif possédant une/des dimension(s) propre(s) mais aussi une/des dimension(s) partagée(s) par les espaces des autres classes.

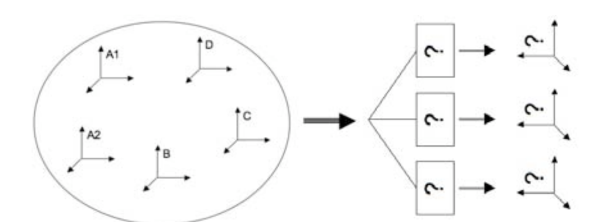


Figure 16 - Hypothèses de la méta-analyse

<u>Expérience</u>

La partie expérimentale de cette étude a été effectuée en deux étapes :

- une première expérience de classification libre avec choix d'un prototype sur le corpus de référence, c'est-à-dire **83 sons** (égalisés en sonie) dont le but est de valider la représentation inter-classes (niveau catégoriel) en définissant les catégories principales et de vérifier/améliorer la représentation intra-classe (niveau continu) en recherchant les meilleurs descripteurs associés aux différentes dimensions perceptives. Cette expérience a été menée auprès de 20 participants (12 hommes / 8 femmes), au moyen d'une interface graphique de classification libre (Matlab, voir Figure 17) permettant de manipuler les sons et de constituer les classes.

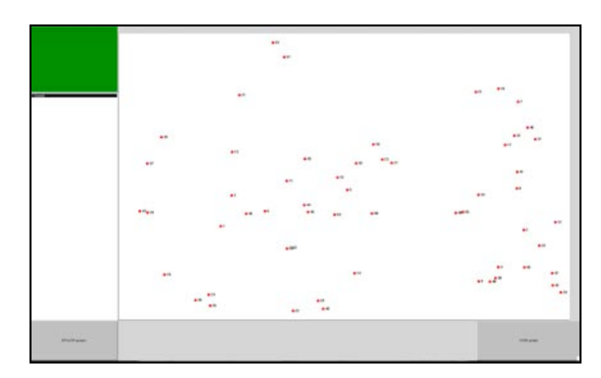


Figure 17 - Interface pour expérience de classification libre (Matlab)

- une seconde expérience de classification orientée, sur la base des résultats (catégories) de l'expérience précédente ; son but est, d'une part, de générer un corpus hétérogène représentant mieux la gamme de variation de chaque classe, et d'autre part, de fournir des données d'apprentissage validées perceptivement pour la mise au point des outils de prédiction (classificateur automatique). Cette expérience a donc été menée sur un nouveau corpus élargi de **150 sons** auprès de 21 personnes (13 hommes / 8 femmes) au moyen d'une interface graphique (psiExp [Smith95]) permettant d'assigner un son à une catégorie pré-définie ; la particularité de ce paradigme étant que chaque classe (sauf la classe 'Autre') est explicitée pour l'utilisateur par un son prototype issu de l'expérience précédente, et non pas par un attribut sémantique comme c'est le cas généralement (voir Figure 18).

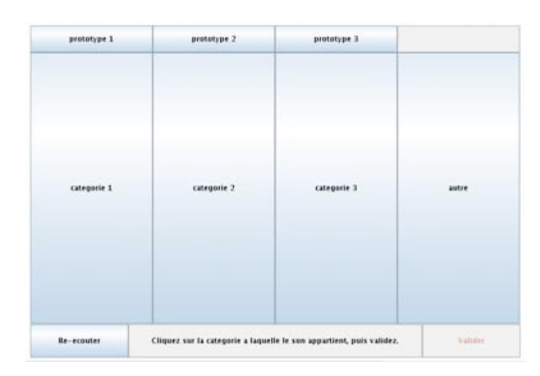


Figure 18 - Interface pour expérience de classification orientée (psiExp)

Les résultats de ces expériences sont décrits au §2.1.4.1.

Les données de la seconde expérience ont de valider perceptivement un corpus élargi définissant le modèle à 3 classes précédemment établi : l'analyse par moyenne et écart-type sur l'ensemble des participants, du nombre de sons placés dans chaque classe permet d'aboutir à une base de **124 sons** : 47 Impact, 50 Moteur, 27 pseudo-Instrument, et donc 26 sons rebutés.

La nouvelle base d'apprentissage issue de l'expérience 2 a comporté suffisamment d'éléments pour concevoir un outil prédictif de classification automatique et en évaluer sa fiabilité :

- à partir d'un modèle de régression logistique multinomiale, modèle probabiliste dont le but est ici de prédire une variable dépendante qualitative et polytomique – les classes – au moyen de variables explicatives continues – les descripteurs – (voir [Taffe04] pour plus de détails). Un premier examen des particularités de chaque classe fait apparaitre la nécessité de prendre en compte à la fois les aspects temporels et spectraux des sons. Plusieurs couples de descripteurs temporels/spectraux sont donc injectés dans le modèle : respectivement {**LAT-** Log-Attack Time, **TI-** Temporal Increase, **TD-** Temporal Decrease, **TC-** Temporal Centroid, **ED-** Effective Duration **EMF-** Energy Modulation Frequency, **EMA-** Energy Modulation Amplitude} / {**SC-** Spectral Centroid *mean*, **SSp-** Spectral Spread *mean*, **SSk-** Spectal Skewness *mean*, **SK-** SpectralKurtosis *mean*, **SSI-** Spectral Slope *mean*, **SD-** Spectral Decrease *mean*, **SR-** Spectral RollOff *mean*, **SV-** Spectal Variation *mean*} (se reporter à [Peeters04a] pour la définition et la formulation de ces descripteurs). La sélection s'effectue au moyen du critère LR- Log-Likelihood Ratio (voir [Taffe04], [Woodcock02] pour plus de détails). Ce critère de sélection fait ressortir le couple ED/SSp – descripteurs caractérisant respectivement, les enveloppes spectrale et temporelle – comme étant le meilleur candidat pour modéliser la taxonomie étudiée (cf. Figure 19)

	LAT	**TI**	**TD**	**TC**	**ED**	**EMF**	**EMA**
SC	167.1	252.7	333.5	373.0	380.5	289.8	101.8
SSp	187.0	257.1	354.6	399.2	407.3	317.5	122.7
SSk	106.1	180.2	314.5	373.1	385.7	235.5	35.7
SK	110.2	188.3	318.9	374.6	386.9	245.1	43.3
SSI	167.1	252.7	333.5	373.0	380.5	289.8	101.8
SD	108.7	187.0	295.9	355.9	369.9	215.8	22.1
SR	98.3	177.9	295.2	353.9	368.3	223.5	43.3
SV	115.6	171.1	306.8	360.9	376.1	224.6	74.0

Figure 19 - Régression logistique – sélection du modèle

- le modèle retenu est validé sur la base d'apprentissage des 124+83 = **207 sons** précédemment constituée selon une méthodologie couramment utilisée :

1. Ré-estimation du modèle (coefficients) sur 70% du corpus ;
2. Calcul de l'erreur sur les 30% restants ;

3. Opération effectuée 100 fois avec sélection aléatoire

Les résultats donnent globalement 5% d'erreurs de prédiction en moyenne (cf.Figure 20)

Minimum recall number	57
Minimum recall percentage	90.5 %
Maximum recall number	63
Maximum recall percentage	100%
Recall number standard deviation	1.3
Mean recall number	60.4
Mean recall percentage	95.9 %
Main recall percentage interval	93.8 % - 97.9 %

Figure 20 - Régression logistique – validation du modèle

Publications dans le cadre du projet Sample Orchestrator

[Minard08a] Minard A., Susini P., Misdariis N., Lemaitre G., McAdams S., Parizet E., « Environmental sound description : comparison and generalization of 4 timbre studies », Computer-Human Interface conference / Sonic Interaction Design workshop, Florence, Italy, 2008.

[Minard08b] Minard A., Susini P., Misdariis N., Lemaitre G., McAdams S., Parizet E., « Two-level description of environmental sounds : comparison and generalization of 4 timbre studies », Acoustics'08, Paris, France, 2008.

[*article soumis*] Misdariis N., Minard A., Susini P., Lemaitre G., McAdams S., Parizet E., « Environmental sound perception: meta-description and modeling based on independant primary studies », EURASIP Journal on Audio, Speech and Music Processing, special issue on Scalable Audio-Content Analysis (SACA), juillet 2009.

Références bibliographiques associées

[Caroll70] Carroll J., Chang J., « Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition », Psychometrica, 35:283-319, 1970.

[Grey77] Grey J. M., « Multidimensional perceptual scaling of musical timbres », J. Acoust. Soc. Am., Vol. 61, No. 5, May 1977.

[Krimphoff94a] Krimphoff J., McAdams S., Winsberg S., « Caractérisation du timbre des sons complexes. II Analyses acoustiques et quantification psychophysique », Journal de Physique Vol. 4, 1994.

[Krumhansl89] Krumhansl C. L., « Why is musical timbre so hard to understand? « in S. Nielzen & O. Olsson (Eds.), Structure and Perception of Electroacoustic Sound and Music, pp. 43-53, Elsevier (Excerpta Medica 846), 1989.

[Lemaitre07a] Lemaitre G., Susini P., Winsberg S., Leinturier B., McAdams S., « The sound quality of car horns: a psychoacoustical study of timbre », *Acta Acoustica*, 93, 457-468, 2007.

[McAdams95a] McAdams S., Winsberg S., Donnadieu S., De Soete G., Krimphoff J., « Perceptual scaling of synthesized musical timbres: common dimensions, specificities, and latent subject classes », Psychological Research, 58, 177-192, 1995.

[McAdams98c] McAdams S., Susini P., Misdariis N., Winsberg S., « Multidimensional characterisation of perceptual and preference judgements of vehicle and environmental noises », *Proceedings of Euro-Noise98*, Munich, Germany, 1998.

[Parizet06] Parizet E., Guyader E., Nosulenko V., « Analysis of car door closing sound quality », *Appl. Acoust.*, 2006.

[Peeters04a] Peeters G. « A large set of audio features for sound description (similarity and classifcation) for the CUIDADO project », http://recherche.ircam.fr/equipes/analyse-synthese/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf, 2004.

[Smith95] Smith, B., « PsiExp: an environment for psychoacoustic experimentation using the IRCAM musical workstation », *in* Society for Music Perception and Cognition conference, University of Berkeley, 1995.

[Susini04b] Susini P., McAdams S., Winsberg S., « Characterizing the sound quality of air-conditioning noise », *Appl. Acoust.*, 2004.

[Taffe04] Taffé P., « Cours de régression logistique », http://www.tesser-pro.org/stat/Cours_regression_logistique.pdf, 2004.

[Winsberg93] Winsberg S., De Soete G., « A latent class approach to fitting the weighted Euclidean model, CLASCAL », Psychometrika, 58, 315-330,1993.

[Woodcock02], Woodcock S. D., « MULTILOGIT.M, implementation of a multinomial logit in MATLAB », http://www.sfu.ca/~swoodcoc/software/software.html, 2002.

1.3.6.2 D-SP2-4-2 – Descripteurs audio de type morphologique pour les sons environnementaux

Responsable : Ircam-PDS

Objectifs

Les profils morphologiques ont pour but de décrire de manière pertinente l'évolution de certains paramètres du son au cours du temps et de proposer une structure d'indexation et de classification prenant en compte ces évolutions. En d'autres termes, cette étude vise essentiellement trois objectifs :

i) la définition de classes de profils morphologiques adaptées aux sons environnementaux,
ii) la conception d'un formalisme pour la description de ces profils, par exemple sous forme symbolique,
iii) l'implémentation d'un modèle de calcul de ces descripteurs temporels.

En outre, on s'intéressera ici plus particulièrement aux **profils dynamiques** (évolution de l'intensité sonore) et aux **profils mélodiques**.

Etat de l'art

La première étape du travail a consisté à intégrer les précédents travaux effectués dans le domaine, notamment ceux réalisés dans le cadre du projet Ecrins.

D'une part, un formalisme de description a été proposé [Deruty01d] à partir d'une l'analyse phenoménologique des sons de l'environnement. Des critères ont été identifiés (profil dynamique, profil mélodique, attaque, espace, texture, etc. …) et ont servi de base à la construction d'une bibliothèque de sons prototypes illustrant ces descripteurs à l'aide d'une double représentation (son de syntèse / son réel). Des propositions de représentations symboliques (icones, lettres, etc. …) ont également été faites pour définir chacun de ces critères.

D'autre part, une validation perceptive de ces idées a été effectuée [Rioux01e], mettant en jeu un paradigme expérimental en trois étapes pour valider les hypothèse de description sur les profils dynamique et mélodique :

A/ classification libre et description (de chaque son) sur des critères morphologiques,
B/ tracé des profils pour chaque son,
C/ reconnaissance des sons à partir de portraits verbaux

Les principaux enseignements à tirer de cette étude sont :

- concernant les stimuli, si les phases A (description + classification) et C (reconnaissance) de l'expérience semblent présenter 11 sons, la phase B (tracés des profils) utilise « des représentants prototypiques de l'un ou l'autre des deux profils (dynamique ou mélodique). » De plus, l'interface expérimentale (HTML) présente 26 sons dont 9 sont associés à des résultats de tracé. Ces 9 sons, synthétiques pour la plupart, sont volontairement difficiles à identifier en termes de source physique : « les sons à source trop facilement identifiable ont été évités (mis à part un son de sirène très marquée) »

- concernant les résultats de la phase A, différents types de descriptions verbales ont été utilisés : vocabulaire shaefferien, description causale, description spatiale, hauteur, description relative à d'autres sons, onomatopées. En outre, il n'y a pas d'analyse approfondie des verbalisations ou des catégorisation effectuées.

- concernant les résultats de la phase B, les tracés de profil ont été moyennés sur l'ensemble des 15 participants, ce qui pose un problème pour les sons présentant des variations relativement rapides (synchronisation entre les participants). Ces tracés moyens ont été corrélés avec la sonie pour les profils dynamiques, et avec la fréquence fondamentale, le centroïde spectral et l'acuité pour les profils mélodiques. La sonie est bien corrélée (> 0.8) pour 5 sons sur 9, le centroïde spectral et l'acuité le sont chacun pour 1 son sur 9, alors que la fondamentale ne l'est jamais – peut-être à cause du choix de la méthode d'extraction. En outre, la résolution temporelle des 4 descripteurs semble trop importante, ce qui peut amener à détériorer significativement les scores de corrélation.

<u>Problématique</u>

La principale différence entre l'étude actuelle et celle réalisée dans le projet Ecrins réside dans la nature même des sons : [Rioux01e] étudie des sons difficilement identifiables alors que c'est globalement le contraire dans le cas présent. En prenant en compte, l'une des principales conclusions de [Rioux01e] : « Malgré la consigne, des descripteurs typiquement causaux ont souvent été utilisés », il apparait donc difficile, dans notre cas, de demander aux participants de décrire la morphologie puisqu'ils seront trop facilement tentés par une description causale.

Par conséquent, pour éviter de trop se disperser dans l'étude de la morphologie des sons, et pour être le plus proche de la notion de « profils », il apparait important d'avoir une phase de classification des sons en quelques profils prototypiques, de sorte à éviter d'avoir une description trop précise des évolutions temporelles (problème qui survient lorsque l'on demande le tracé de chaque son). Par ailleurs, il apparait également judicieux, dès cette première phase, de demander aux participants de tracer les profils de leurs propres classes de sons, de manière à les obliger à se concentrer sur les principaux « traits » morphologiques communs aux sons de chacune des classes.

<u>Expérience</u>

L'étude expérimentale de la perception des profils morphologiques a été effectuée sur un sous-ensemble (55 sons) de la base de données des sons environnementaux des expériences menées dans le SP2-5 (cf. expérience d'identification); elle comporte finalement deux étapes :

- une première expérience en deux phases : classification libre sur des critères morphologiques et tracé des profils correspondant aux classes identifiées. L'intérêt de ce paradigme réside dans le fait que les participants, sachant qu'il devront tracer les profils de leurs classes, sont "forcés" à effectuer la classification selon des critères morphologiques ; l'inconvénient majeur étant la difficulté de traitement des tracés de différents participants qui n'ont à priori pas les mêmes classes.

Cette étape est réalisée avec 19 participants (14 hommes / 5 femmes), tous experts dans le domaine de la musique et du son (musiciens, ingénieurs du son, personnes travaillant régulièrement dans des environnements audionumériques, etc. …). Elle est effectuée au moyen d'une interface graphique de classification libre (Matlab, voir Figure 17) permettant de constituer les classes et d'une interface temps-réel (Max/MSP) reliée à une tablette WACOM Intuos [Wacom] permettant de recueillir les données morphologiques ; la communication entre les deux environnements s'effectuent grâce au protocole OSC (OpenSound Control). La procédure est répétée successivement pour les profils dynamiques et les profils mélodiques (voir .Figure 21).

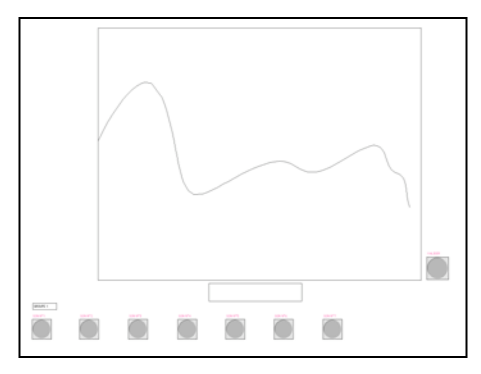


Figure 21 - Interface de description des profils morphologiques (Max/MSP + Wacom)

- une seconde expérience, sur la base de l'analyse des résultats précédents de classification et l'extraction des classes 'moyennes', qui consiste à demander aux participants de tracer le profil correspondant à chacune de ces classes, et de juger de la pertinence des classes proposées sur une échelle de allant de 0 (« pas du tout pertinent ») à 10 (« tout à fait pertinent »). Les résultats permettront principalement d'obtenir des éléments de symbolisme correspondant aux classes identifiées, afin d'associer, à terme, un symbole à chacune d'elle.

Cette étape est réalisée avec 19 participants (12 hommes / 5 femmes), tous experts dans le domaine de la musique et du son (musiciens, ingénieurs du son, personnes travaillant régulièrement dans des environnements audionumériques, etc. …). Elle est effectuée au moyen d'une interface temps-réel (Max/MSP) reliée à une tablette WACOM permettant de jouer les sons et recueillir les données morphologiques (voir Figure 21). La procédure ne concerne que les profils dynamiques.

Les résultats obtenus sont donnés au §2.1.4.2.

Perspectives

Concernant les profils dynamiques les perspectives peuvent être les suivantes : à court terme, affiner la proposition de symboles basés sur le tracé des participants et identifier les prédicteurs efficaces de ces 6 classes ; à moyen terme, envisager une troisième expérience permettant de valider les éléments symboliques établis, sur un corpus sonore élargi.

Concernant les profils mélodiques, il faut avant tout s'interroger sur la pertinence d'appliquer une telle notion au cas de sons environnementaux et, le cas échéant, adapter la méthodologie expérimentale afin d'obtenir des résultats comparables à ceux obtenus pour les profils dynamiques.

Références bibliographiques associées

[Deruty01d] Deruty E., « Les descripteurs morphologiques des sons : descripteurs2.doc », rapport interne ECRINS, mai 2001.

[Rioux01e] Rioux V., « Projet ECRINS / validation expérimentale phase I: descripteurs morphologiques », rapport interne ECRINS, novembre 2001.

[Wacom], Wacom Pen Tablet, http://www.wacom.com/intuos/

1.3.7 SP2-5 – Classification des sons environnementaux

1.3.7.1 D-SP2-5-1 – Définition d'un corpus représentatif et sélection des classes pertinentes

Responsable : Ircam-PDS

Objectifs

La dimension quasi-infinie du champ couvert par les sons dits "environnementaux" nécessite, dans un premier temps, de travailler sur une representation réduite mais significative de cet univers sonore, du moins si l'on aborde le problème du point de vue expérimental. La constitution de cette base de données représentative constitue donc un

passage obligé et est élaborée grâce à l'expertise d'utilisateurs experts, habitués à manipuler différentes typologies sonores (collaboration avec Univers Sons).

Résultats

Le corpus de départ est fourni par Univers Sons le 27/09/2007 et se compose de **368 sons** répartis en **46 classes ("**SOR_database"). La classification d'usage associée comporte un seul niveau de hiérarchie, et les classes sont définies par des critères multiples, mélangeant acoustique/événement et catégories d'objets. Ces différents critères, même s'ils sont mélangés recoupent les différents types de classification introduits dans le D-SP2-6 (acoustique, évènementiel, sémantique). Parmi les critères définissant ces 46 classes, on trouve des critères de type évènementiel (cause physique ou matériau), acoustique (trait acoustique ou évolution temporelle) ou sémantique (voir Figure 22). Par ailleurs, il apparaît clairement que des sons d'une classe correspondant à un type de description donné pourraient également être placés dans une autre des classes définies, selon un autre type de description.

Sur cette base, un premier traitement à consister à unifier l'ensemble (fréquences d'échantillonnage, formats d'encodage, etc. …) et à segmenter certains fichiers qui comportaient une séquence de plusieurs sources distinctes ; ces opérations ont été effectuées dans le cadre du SP2-2 et ont abouti à un corpus modifié de **495 sons** et toujours **46 classes ("**SOR_database_segmented").

Une deuxième passe de traitement de la base de données sonores initiale a ensuite consisté à confronter ces classes d'usage avec les résultats de travaux plus théoriques sur la perception des sons du quotidien afin de replacer ces échantillons prototypes dans des organigrammes de référence proposant une taxonomie globalement plus pertinente du point de vue perceptif (c'est-a-dire qui fait statistiquement sens pour un ensemble d'utilisateurs/auditeurs) et mettant en évidence plusieurs niveaux de catégorisation des sons environnementaux. En s'appuyant donc essentiellement sur les travaux originaux de Gaver [Gaver93] et ceux – plus récents – accomplis dans le cadre du projet européen CLOSED [Closed] par Houix&al. [Houix07a] [Houix07b], cette phase a eu deux actions :

• conformément aux conclusions de [Houix07b] (également abordées dans SP2-6), distinguer les différents niveaux de similarité des sons selon le type d'information contenu dans le signal sonore et perçu par l'auditeur : principalement /Acoustique/, /Evénementiel (Causal)/ ou /Sémantique/ : en effectuant certaines associations – notamment Evenement+Materiau et Acoustique+Séquence -–, on aboutit donc à un redécoupage de la base comme suit : 36 critères évènementiels, 7 critères de acoustiques et 4 critères sémantiques (voir Figure 22).

Evènementiel :
Crackle : applause, fire, hail
Gaz, air : airpumps, release, grammaphone (?), match (?)
Fire (allumage) : fireworks, torch…
Bubbles :
Buzz : buzz, bees, flies
Chimes (-> matériau ? voire sémantique) : divers bells
Clicks : soft clicks divers objets (switches, belt, metronome, clap, clock)
Creaks : squeaks divers + firework, guiro scotch, switch (molette ?)
Explosions : divers, mauvaise qualité
Fizzing : soda cans…
Hit : punches, kick, « pauke » (?)
Humming : moteurs, snare roll (?)
Mud, glue : half-liquid splashes (blood, sloshes, sneezes)
Pschht : air, cymbal, tray => ????
Scraping : knives
Scratch : zips, nails, scotch
Shots : guns, fireworks, (largeblocks ?????)
Springs : divers ressorts

Squeaks :
Long : gasp,balloons, door, firework
Short : bird, squeals…
Tinkles (-> séquentiels ?) : coins, chimes
Vibrating (-> séquentiels) : motors, pumps
Waters :liquids
Large : riffs, seas, lakes
Small : bottle, drops, soft streams
Whir : air conditionners, freezer, microwave (sea flood ?)
Woosh : whip, wipes
Matériau :
Metal : divers impacts d'objet de divers tailles
Glass : divers bris + soft impacts
Ceramic (proche glass voire metal) : divers vaisselle
Brick : impacts bricks, tiles, cement
Paper : froissements, mouvements
Plastic : boxes, bag
Stone : impacts + frottements
Wood : divers impacts + torsions
Séquentiel (pattern temporel) :
Régulier : sons divers (alarm, applause gun helicopter)
Accéleration : coins, vibrato (triangle)
Random : brushing teeth, crickets, morse, switch
Sweep : profils mélodiques up ou down
Sémantique :
Machine (-> séq. ou évèn. ?) : typewritters, phone dial, telegraphs, printers
Electronic : electrocution, high voltage, laser, radio (?)
hubbub : crowdy ambiances
rumble : space ambiances (sourds -> specif acous ?)
Spécif. acoustique :
Low, blunt , divers impacts sourds
Reverse (-> séquentiel ?): vocal, piano
Simple waves : pure tones : phones, radio, beeps…

Figure 22 - Corpus sonore représentatif (Univers Sons)

Taxonomie d'usage vs. 1er niveau de structure (Acoustique/Evènementiel/Sémantique/…)

• étant donné l'intérêt porté sur le niveau /Evènementiel/, se focaliser sur la perception des sources, c'est-à-dire de la cause physique du son, et proposer, *à priori*, une ré-organisation de la base de données initiale en accord avec la structure interne de ce niveau issue des travaux de [Gaver93] et des validations effectuées par [Houix07b].

Les résultats de cette tâche sont décrits au §2.1.5.1.

1.3.7.2 D-SP2-5-2 – Organisation des classes pertinentes

Responsable : Ircam-PDS

<u>Objectifs</u>

L'objectif global de cette tache est de fournir des éléments d'intégration pour le SP2-2, sous la forme d'un corpus classifié et validé perceptivement, afin de tester les méthodes de classification automatique. Une étude expérimentale est donc menée sur la base des hypothèses résultant du SP2-5-1 (taxonomie causale).

<u>Expérience</u>

L'expérience consiste en une identification de la cause physique du son, c'est-à-dire du type d'action physique qui l'a produit, (indépendamment de l'objet, de sa forme et du matériau qui le constitue), selon les deux niveaux de description représentés sur la Figure 54 : 1er niveau (jaune), 2ème niveau (bleu). Le test est réalisé auprès de 20 participants (9 hommes / 11

femmes) au moyen d'une interface graphique de classification orientée (psiExp [Smith95]) donnant accès séquentiellement aux 2 niveaux de la taxonomie (voir Figure 23). Principalement pour des raisons de temps de passation, le corpus d'étude utilisé est un sous-ensemble du corpus de travail issu du SP2-5-1 ("SOR_database_structured") : il est constitué de **213 sons** respectant au mieux les distributions initiales : 33– Liquide (li), 92– Solide (so), 49– Gaz (ga), 10– Electrique (el), 12– Electronique (sy), 17– Machine (mo).

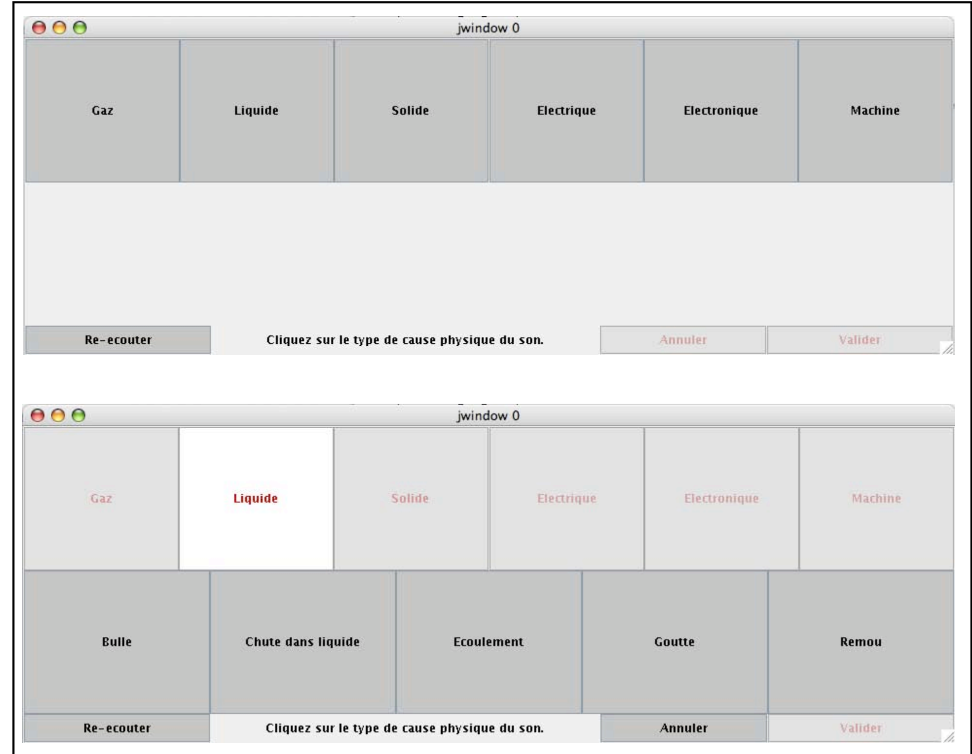


Figure 23 - Interface pour expérience de classification orientée (psiExp)

Description du 1er niveau (haut) et du 2ème niveau (bas)

<u>Résultats</u>

Les données de l'expérience ont été analysées en terme de matrice de confusion (en %), qui représente le taux d'identification de chaque classe en calculant – en moyenne sur les participants – le nombre de sons correctement assignés à la classe à laquelle il appartient théoriquement (classes « théoriques » définies dans le SP2-5-1). Cette analyse s'est effectuée suivant les deux niveaux de la structure de classification :

- 1er niveau (jaune sur Figure 54) :

		Classes identifiées par les participants					
		Gaz	Liquide	Solide	Electrique	Electronique	Machine
Classes théoriques	Gaz (49)	67.9	0.8	15.8	1.5	2.8	11.3
	Liquide (33)	3.8	89.3	1.8	1.4	2.4	1.3
	Solide (92)	1.9	0.4	89.9	0.9	2.1	4.9
	Electrique (10)	1.1	0.0	0.0	61.6	25.3	12.1
	Electronique (12)	0.0	0.0	4.8	9.6	80.3	5.3
	Machine (17)	15.5	1.2	1.9	8.4	2.5	70.6

Figure 24 - Matrice de confusion (en %) – 1er niveau

- 2ème niveau (bleu sur Figure 54) ; <u>nb.</u> : les classes 'glue' et 'ressorts' ont été rajoutées *à posteriori*, pour des raisons expérimentales) :

	crepitement	explosion	sifflement	souffle	bulle	liquide	ecoulement	goutte	remou	glue	bris	choc	clic	ecrasement	fermeture	froissement	frottement	grincement	roulement	tintement	ressorts	Electrique	Electronique	Machine	
crepitement	65.0	7.1	2.0	11.9	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	5.0	0.0	0.0	3.8	0.0	0.0	0.0	0.0	0.0	0.0	2.1	
explosion	0.0	51.8	2.0	5.5	0.0	0.0	0.0	0.0	0.0	0.0	0.7	1.2	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	
sifflement	0.0	0.3	64.0	14.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2	0.5	0.0	0.2	0.0	0.5	0.0	0.3	
souffle	10.8	3.8	2.0	35.0	0.4	0.0	0.8	0.0	11.4	0.0	0.0	0.0	1.0	0.0	0.0	0.0	2.7	0.5	0.0	0.0	2.5	0.5	0.0	12.6	
bulle	0.8	0.0	0.0	0.2	40.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
chute dans liquide	0.0	0.0	0.0	0.2	4.3	85.0	5.8	32.5	5.0	41.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	
ecoulement	0.0	0.0	0.0	1.2	19.3	0.0	89.2	0.0	25.0	6.7	0.0	0.0	0.0	0.0	0.0	0.0	1.2	0.0	0.0	0.0	2.5	0.0	0.0	0.9	
goutte	0.8	0.0	0.0	0.0	0.4	0.0	0.0	62.5	0.0	6.7	0.0	1.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
remous	0.8	0.0	0.0	0.0	23.6	15.0	4.2	0.0	56.4	23.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.5	0.0	0.0	0.0	
glue	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
bris	0.8	0.3	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	89.3	13.1	0.0	10.0	0.0	0.0	0.8	0.0	0.0	5.7	0.0	0.0	0.0	0.0	
choc	0.8	14.1	0.0	3.1	0.0	0.0	0.0	0.0	0.0	1.7	7.9	70.6	38.0	10.0	47.5	0.0	0.0	0.0	38.0	23.0	15.0	0.0	0.4	0.0	
clic	1.7	1.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	1.2	22.5	6.7	0.0	0.0	0.8	0.5	1.5	15.9	0.0	0.0	0.0	0.3	
ecrasement	5.0	1.2	2.0	0.7	0.0	0.0	0.0	0.0	0.0	13.3	1.4	5.6	0.5	26.7	5.0	11.2	2.3	0.5	5.0	3.0	0.0	0.0	0.0	0.0	
fermeture	0.0	0.3	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.5	0.0	45.0	0.0	13.5	6.0	0.0	0.0	0.0	0.0	0.0	0.0	
froissement	4.2	0.6	0.0	1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	16.7	0.0	75.0	0.4	0.0	0.0	0.2	0.0	0.0	0.0	0.0	
frottement	3.3	2.4	1.0	8.6	0.0	0.0	0.0	0.0	0.0	3.3	0.0	0.0	8.5	6.7	0.0	11.2	46.9	16.5	1.5	0.7	2.5	0.0	0.0	0.6	
grincement	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.7	0.0	0.0	5.4	63.5	0.5	0.0	0.0	0.0	0.0	0.0	
roulement	1.7	0.0	0.0	1.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2	0.5	5.0	0.0	1.2	11.2	0.0	46.0	3.5	0.0	0.0	0.0	0.6	
tintement	0.0	0.9	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.6	7.5	3.3	2.5	0.0	0.4	1.0	7.5	40.9	2.5	0.0	4.6	0.3	
ressorts	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
Electrique	0.8	1.5	4.0	1.4	3.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.8	0.0	0.0	1.1	20.0	62.0	11.2	8.8	
Electronique	0.0	2.4	6.0	2.9	4.3	0.0	0.0	5.0	0.7	1.7	0.0	0.0	3.0	0.0	0.0	0.0	4.6	0.5	0.0	2.2	20.0	24.5	78.8	2.4	
Machine	3.3	12.4	15.0	11.0	2.1	0.0	0.0	0.0	0.7	1.7	0.0	0.0	9.5	8.3	0.0	1.2	4.2	10.5	0.0	3.7	32.5	12.5	5.0	70.6	
nb_sons	**6**	**17**	**5**	**21**	**14**	**1**	**6**	**2**	**7**	**3**	**7**	**8**	**10**	**3**	**2**	**4**	**13**	**10**	**10**	**23**	**2**	**10**	**12**	**17**	**213**

Figure 25 Matrice de confusion (en %) – 2ème niveau

Classes théoriques (vertical) vs. classes identifiées par participants (horizontal)

Les conclusions issues de ces travaux sont présentées au §2.1.5.2.

Perspectives

La poursuite de ce travail peut s'envisager à deux niveaux :

. d'un point de vue théorique, améliorer la structure de classification en affinant notamment la description de certaines de ses composantes (par exemple, Gaz ou Liquide) ;
. d'un point de vue expérimental, valider ces nouveaux résultats en tentant de résoudre la principale difficulté rencontrée : la taille importante du corpus qui s'avère incompatible avec la durée une expérience conventionnelle. Une solution possible à ce problème – qui a déjà commencée à être étudiée – serait d'utiliser des méthodes d'apprentissage automatique supervisées dans le but de concevoir des paradigmes expérimentaux adaptatifs, c'est-à-dire évoluant avec les décisions prises par le participant au cours du test.

Références bibliographiques associées

[Closed] « Closing The Loop of Sound Evaluation and Design », projet européen FP6-NEST-PATH No: 29085, 2006-2009, http://closed.ircam.fr/

[Gaver93] Gaver W. W., « How do we hear in the world? Explorations in ecological acoustics », Ecological Psychology, 5(4):285–313, 1993.

[Houix07a] Houix O., Lemaitre G., Misdariis N., Susini P., Franinovic K., Hug D., Otten J., Scott J., Visell Y., Devallez D., Fontana F., Papetti S., Polotti P., Rocchesso D., « Everyday sound classification. Part 1 : Sound perception, interaction and synthesis », Deliberable 4.1 of the CLOSED project, 2007, http://closed.ircam.fr/uploads/media/closed_deliverable_4.1_public.pdf

[Houix07b] Houix O., Lemaitre G., Misdariis N., Susini P., « Everyday sound classification. Part 2 : Experimental classification of everyday sounds », Deliberable 4.1 of the CLOSED project, 2007, http://closed.ircam.fr/uploads/media/closed_deliverable_4.1_part2_public.pdf

[Smith95] Smith, B., « PsiExp: an environment for psychoacoustic experimentation using the IRCAM musical workstation », *in* Society for Music Perception and Cognition conference, University of Berkeley, 1995.

1.3.8 SP2-6 – Caractérisation sémantique des sons environnementaux

1.3.8.1 D-SP2-6-1 – Description sémantique des sons

Responsable : Ircam-PDS

Objectifs

L'objectif de cette tâche est d'effectuer un travail exploratoire sur les descripteurs permettant de décrire d'un point de vue sémantique des classes de sons (sons de pas …). Cette tâche est alimentée par la définition de classes de sons environnementaux qui ont un sens du point de vue d'utilisateurs experts (designer sonore, bruiteur, ...), c'est-à-dire par une partie des données fournies par Univers Sons et formalisées dans SP2-5-1 (voir Figure 22). Le but à atteindre est donc la description des éléments de ce corpus en termes de descripteurs sémantiques pertinents et discriminants.

Résultats

Les travaux en perception et cognition auditive sur la classification des sons du quotidien abordés dans le cadre du projet CLOSED [Closed], se basant sur la littérature dans ce domaine, ont montré que des auditeurs utilisaient différents types de similarité pour catégoriser des sons :

i) à partir d'attributs perceptifs directement liés aux propriétés acoustiques,
ii) sur la base de la cause physique du son,
iii) sur la base d'une abstraction de propriétés associées à l'appartenance à une catégorie d'objets.

Dans le cadre d'une base de données sonores, nous proposons différentes structures associées à chacune de ces similarités (acoustique, événement, catégories sémantiques – voir Figure 26) :

- Les relations de similarité acoustiques peuvent être directement estimées sur la base de descripteurs calculés sur le signal contrairement aux deux autres types de similarité.

- La structure représentant les relations entre type d'événements peut être représentée par un arbre hiérarchique avec plusieurs niveaux d'inclusions (en nombre limité), partant de la distinction entre Liquide / Gaz / Solide, avec dans chacune de ces classes des sous classes (p. ex., pour les solides : impact, friction, écrasement). Cette structure implique un ou plusieurs labels associés à chaque son pour le situer dans une ou plusieurs classes.
- Les relations entre catégories d'objets sont quant à elles plus complexes et peuvent être représentées sous la forme d'un réseau sémantique tel que Wordnet [Wordnet], utilisé notamment dans le projet AudioClas [Cano04], incluant des relations transversales entre catégories, notamment :
  . synonymie
  . antonymie
  . hyponymie (« sorte de …»)
  . méronymie (« partie de …», « sorte de… », ...)

	Description	**Organisation**
Acoustique	Timbre, morphologie	Espace multidimensionnel de timbre
Evènementielle	Cause et matériau	Structure hiérarchique (squelette)
Sémantique	Objet, lieu, fonction…	Réseau sémantique (hiérarchie + liens transversaux*)

*Figure 26 - Synthèse des différents type de description d'un son (* liens transversaux : synonymie, antonymie, hyponymie, méronymie, …)*

Plus précisément, le principe du réseau sémantique Wordnet repose sur la représentation d'un concept comme, d'une part, un ensemble de synonymes et d'antonymes (relations lexicales), et d'autre part, un ensemble d'hyponymes/hyperonymes et de méronymes/holonymes (relations sémantiques). Un son donné pourra donc avoir plusieurs types de description associés, et, en fonction de ceux-ci, apparaîtra dans les organisations correspondantes au milieu de voisins directs différents d'une organisation à l'autre. Une illustration de cette construction et de sa représentation sous Treebolic [Treebolic, Cano04] est donnée avec le mot '***step***' dans la Figure 27.

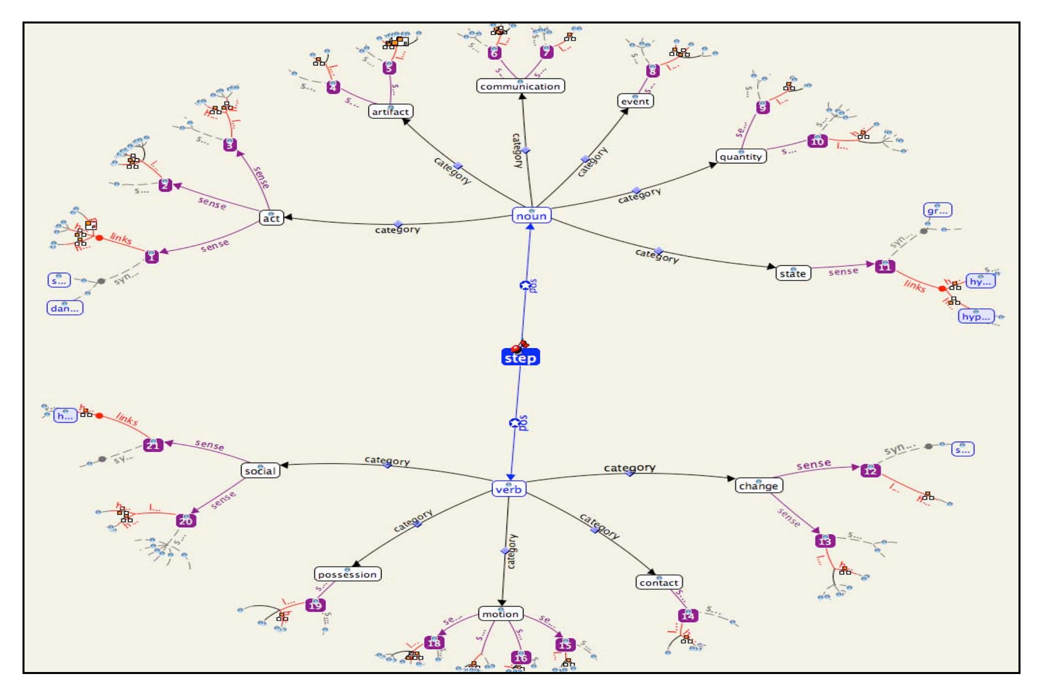


*Figure 27 - Représentation du réseau sémantique associé à '**step**' sous Treebolic*

<u>Références bibliographiques associées</u>

[Cano04] Cano P., Koppenberger M. Herrera P., Celma O., « Sound Effect Taxonomy Management in Production Environments », proceedings of 25th International AES Conference London, UK, 2004, http://audioclas.iua.upf.edu/

[Closed] « Closing The Loop of Sound Evaluation and Design », projet européen FP6-NEST-PATH No: 29085, 2006-2009, http://closed.ircam.fr/

[Houix07a] Houix O., Lemaitre G., Misdariis N., Susini P., Franinovic K., Hug D., Otten J., Scott J., Visell Y., Devallez D., Fontana F., Papetti S., Polotti P., Rocchesso D., « Everyday sound classification. Part 1 : Sound perception, interaction and synthesis », Deliberable 4.1 of the CLOSED project, 2007, http://closed.ircam.fr/uploads/media/closed_deliverable_4.1_public.pdf

[Houix07b] Houix O., Lemaitre G., Misdariis N., Susini P., « Everyday sound classification. Part 2 : Experimental classification of everyday sounds », Deliberable 4.1 of the CLOSED project, 2007, http://closed.ircam.fr/uploads/media/closed_deliverable_4.1_part2_public.pdf

[Wordnet] Miller G. A., « Base de données lexicales », http://wordnet.princeton.edu/

[Treebolic] Bou B., « Application Web pour Wordnet », http://jws-champo.ac-toulouse.fr:8080/wordnet-treebolic/

1.3.8.2 D-SP2-6-2 – Description sémantique des classes

Responsable : Ircam-PDS

L'objectif de cette tâche était d'effectuer un <u>travail exploratoire</u> sur les descripteurs permettant de décrire d'un point de vue sémantique les sons (type de matériau, mouvement, ...). Faute de temps, cette partie n'a pu être mise en oeuvre dans le cadre du projet ; elle n'a donc produit aucune connaissance et, *a fortiori*, aucun résultat significatif dans ce domaine.

1.3.9 SP2-7 – Recherche, navigation et composition interactive fondée sur des bases de sons

La synthèse concaténative par corpus utilise une base de données de sons enregistrés, et un algorithme de sélection d'unités qui choisit les segments de la base de données qui conviennent le mieux pour la séquence musicale que l'on souhaite synthétiser par concaténation [SCHWARZ07a][SCHWARZ07c]. La sélection est fondée sur les caractéristiques de l'enregistrement, qui sont obtenues par analyse du signal et correspondent par exemple à la hauteur, à l'énergie ou au spectre. Les méthodes de synthèse musicale habituelles sont

fondées sur un modèle du signal sonore, mais il est très difficile d'établir un modèle qui préserverait la totalité des détails et de la finesse du son. En revanche, la synthèse concaténative, qui utilise des enregistrements réels, préserve ces détails.

La mise en œuvre en temps réel de cette nouvelle approche de la synthèse sonore par le système CataRT permet une exploration interactive d'une base sonore et une composition granulaire ciblée par des caractéristiques sonores précises, et permet aux compositeurs et musiciens d'atteindre de nouvelles sonorités, et aux designers sonore de rapidement explorer un corpus sonore constitué par un grand nombre d´enregistrements.

Le système modulaire CataRT [SCHWARZ07b][Schwarz08a] de synthèse concaténative par corpus interactive en temps réel est implémenté dans Max/MSP avec les extensions FTM, Gabor et MnM. CataRT est utilisé dans des contextes musicaux de composition, de performance, et d'installations variés. L'interaction repose sur une interface simple consistant en l'affichage d'une projection 2D de l'espace de descripteurs, et une navigation avec la souris, où les grains sont sélectionnés et joués par proximité géométrique.

1.3.9.1 D-SP2-7-3 – Application et modules finaux de navigation et synthèse par corpus

Responsable : Ircam-IMTR

Les travaux réalisés pour le projet Sample Orchestrator, détaillés dans les sections suivantes, s'organisent autour du développement de CataRT comme application prototype pour expérimenter des paradigmes de recherche et création sonore par navigation dans des grandes collections sonores, ainsi que des modules, bibliothèques de fonctions, et sous-systèmes, qui ont été développés et testés dans CataRT, mais qui sont intégrables et utilisables dans d'autres applications. Les points suivants ont été développés :

- Librarie *DistLib* (1.3.9.2) contenant des algorithmes de recherche efficaces dans des grandes collections de données, notamment par arbres de recherche binaires (k*D-Trees*), et des définitions de distances sur des données symboliques hiérarchiques
- Librarie *VisLib* pour le support de la visualisation de données multidimensionnelles (1.3.9.2) en utilisant des méthodes de réduction de dimensionnalité tel que multi-dimensional scaling (MDS), principal component analysis (PCA) avec l'intégration de poids par dimension, linear discriminant analysis (LDA) et *mapping* linéaire
- Avancement de l'architecture et des fonctionnalités de CataRT comme modèle d'une application de navigation dans des grandes bases sonores, et de sa documentation (1.3.9.4 et cf. rapport M22 annexes 1 et 2)
    - Intégration des nouvelles fonctionnalités développées en libraries et autres modules dans CataRT: *DistLib, idesc,* et *VisLib*
    - Développement de nouveaux descripteurs spectraux
    - Nouvelle architecture de l'interface graphique pour la recherche de sons et la navigation dans des grandes collections sonores (1.3.9.5)
    - Méthodes d'interaction avec la continuation de boucles rythmiques
- Version *standalone* simplifié de CataRT, servant de demonstrateur
- Modélisation de la continuation de boucles rythmiques (1.3.9.6, voir aussi SP3-4 – Variation interactive de boucles rythmiques) : ces recherches ont été effectués au sein de l'application CataRT et ont résulté dans des parties de la librairie *DistLib* et des nouvelles fonctionnalités de CataRT.

1.3.9.2 Librarie DistLib: Algorithmes de recherche efficaces dans des grandes collections de données multidimensionnelles

La bibliothèque de fonctions *DistLib* fournit des fonctions et structures de données pour le calcul de distances et la recherche d'éléments dans des espaces de données

multidimensionnelles. Cette librarie n'a pas de dépendances envers d'autres libraries et peut s'intégrer dans n'importe quelle application. Elle a été testé dans l'application prototype de recherche navigation dans des grandes collections sonores *CataRT*. Des tests de performance des algorithmes ont été effectués et sont documentés en bas.

Les algorithmes qui sont unifiés dans *DistLib* sont :

- le calcul de distances avec poids (distance de Mahalanobis)
- le calcul de distance de continuation avec poids (cf. 1.3.9.6)
- des arbres de recherche binaires (k*D-Trees*),
- des définitions de distances sur des données symboliques hiérarchiques par matrice de distance

La documentation détaillé de l'API se trouve dans le rapport M22 et les résultats de mesures du temps de calcul en section 2.

Une amélioration future du kD-Tree consiste en l'utilisation d'un découpage optimal pour la décomposition de l'espace des points perpendiculaire aux composantes principales par une analyse par PCA. D'haes et al (2004) montrent qu'ainsi l'arbre de recherche devient optimal. Un autre avantage est que le fléau de la dimensionnalité est réduit, car seulement les dimensions intrinsèques de l'espace de recherche, c.a.d. indépendantes des autres, comptent.

1.3.9.3 Librarie VisLib *pour la Visualisation de données multidimensionnelles*

La navigation interactive dans des espaces de données multidimensionnelles demande soit une exploration des dimensions un-à-un, soit une réduction des multiples dimensions sur deux ou trois pour permettre leur affichage: Des méthodes de réduction de dimensionnalité tel que *multi-dimensional scaling (MDS)*, *principal component analysis (PCA)* avec l'intégration interactive de poids par dimension, le *mapping* linéaire (matmap), et, si des labels de classes sont disponible, la *linear discriminant analysis* (LDA), peuvent aider à rendre accessible les espaces de haute dimensionnalité pour la navigation interactive. Certains de ces algorithmes sont groupés dans la librarie *VisLib*, sans dépendances envers d'autres libraries et donc facilement intégrable dans n'importe quelle application

Une simulation de ce procédé en utilisant l'algorithme *Metric MDS* de Matlab avec une norme de distance Euclidienne avec normalisation montrait l'intérêt de MDS.

Ainsi a été implémenté l'algorithme *Hybrid MDS*, qui est une version optimisée itérative de MDS avec des itérations de compléxité sub-linéaire basé sur recherche des plus proches voisins (kNN) [Morrison, Ross, Chalmers, InfoVis 2002, 2003]. Cet algorithme se base sur une modélisation masse-ressorts. L'approche itérative du placement des points vers la solution optimale peut déjà être visualisé ce qui permèt à l'utilisateur d'interrompre l'algorithme dès que la solution intermédiaire est suffisante pour la tache de navigation requise. Pour la recherche des plus proches voisins, nous disposons déjà d'une implémentation optimale en temps logarithmique par le kD-Tree de la librarie *DIstLib*.

L'algorithme procède en trois phases :

1. Initialisation: Un échantillon arbitraire de $n_{samp} = \sqrt{,N}$ points est placé à l'aide d'un modèle masse-ressort complet, afin de commencer avec un placement convenable.

2. Interpolation: Les points restants sont placés autour de leur plus proche voisin dans les données.

3. Itération: Tous les points sont arrangés à l'aide d'un modèle masse-ressort réduit, qui ne garde que n_{ngb} liens vers les plus proches voisins dans l'espace d'affichage, et n_{rand} liens sont choisis arbitrairement à chaque itération afin de faire suivre la forme globale.

Le choix de n_{samp} a comme conséquence que chaque itération dans la phase d'initialisation est linéaire, puisque un system complet prend $O(n^{2,samp})=O(N)$. En utilisant notre *k*D-tree

nous pouvons réduire la complexité de la phase d'interpolation à $O(N\log N)$. Les itérations finales sont sublinéaires, avec un nombre constant $n_{ngb}+n_{rand}$ de liens à évaluer, et quelques itérations seulement suffisent pour la convergence.

Nous avons ajouté au modèle masse-ressort une force de répulsion, qui sert déjà à l'amélioration de la visualisation (voir plus bas).

1.3.9.4 Avancement de l'architecture et des fonctionnalités de CataRT

CataRT est un modèle d'une application de navigation dans des espaces sonores, et un cadre de test pour des nouveaux paradigmes de visualisation et recherche dans des grandes bases de sons. Les avancées réalisées pour la version finale du prototype concernent principalement des adaptations du système à la recherche de sons dans des grandes collections sonores (sound effects), et de sa documentation (cf. rapport M22 Annexes 1 et 2). Ils sont groupés par fonctionnalités et détaillés dans le suivant.

L'avancée de l'architecture logicielle concerne principalement la visualisation (décrit en détail dans la section 1.3.9.5) et une nouvelle architecture d'importation et analyse des sons : elle comporte la définition d'interfaces pour modules d'analyse en descripteurs, de modélisation temporelle, et de segmentation, qui se déclarent au modèle de donnée, et peuvent donc facilement être ajoutés, changés et comparés.

Définition des descripteurs sonores

En ce qui concerne la définition des descripteurs, le système à été rendu plus flexible en distinguant différents types, mais de permettre une présentation claire de la liste des descripteurs par groupement et élimination de descripteurs internes :

- Différent types de descripteurs peuvent maintenant être distingués :
    - int, float, symbol, vector, classe
    - Les classes produits par les modules d'indexation de SP2-2 et SP2-3 sont sans hierarchie, et représentés comme symbol et probabilités pour chaque classe.
- Visibilité et groupement dans l'interface
- Représentation d'une taxonomie de classes

Analyse et segmentation

L'analyse en descripteurs et la segmentation ont progressé par la définition de nouveaux algorithmes de descripteurs spectrales (cf. 1.3.9.6) et de segmentation, et de nouveaux formats d'importation, en outre de la refonte de l'architecture logicielle décrite en haut, qui permet d'intégrer facilement d'autres algorithmes, comme par exemple les descripteurs de SP2-1.

- Formats d'importation de la segmentation
    - Fichiers de markers SDIF, text, labels
    - Markers dans AIFF, WAV
- Chargement des sons dans le format de compression sans perte FLAC (*free lossless audio codec)*.
- Utilisation du format SDIF pour les descripteurs de bas-niveau et pour la persistence des corpus (via Mubu du SP3)
- Chargement de descripteurs globaux externes par fichier texte
    - Par fichier son
    - Par repertoire

*V*isualisation

Le sous-système de visualisation à progressé notamment par l'amélioration de l'architecture et du modèle graphique décrite en détail plus bas, et des nouveaux modes d'affichage, voir 1.3.9.5 pour plus de détails.

- Espace de coordonnées séparés en *model, world, device*
  - transformations affines ou mapping entre espaces
  - 20x plus rapide que la version précedente, non-vectorielle
  - permet une interpolation animée des points au changement des axes
- labels par unité, par groupe (sur centre statistique)
- Mode *tile* de mosaiques de sous-espaces
- Recherche par catégories/classes
  - Affichage d'un menu des classes par classification (matériau, onomatopées, repertoires de la base de sons)
  - « solo » cumulatif d'une classe, les points des autres classes restent visibles en gris

<u>Synthèse et sélection</u>

Le contrôle de la sélection peut s'effectuer maintenant par le séquençage d'un fichier de sons analysé (musaicing), ou par un fichier MIDI donnée. Un nouveau module de synthèse à été développé qui fonctionne par lecture directe de disque, ce qui permet l'utilisation de très grands corpus, sans charger les sons en memoire.

<u>Documentation</u>

La documentation à été amelioré, validé par un utilisateur qui ne connaissait pas le système, et structuré dans ces deux documents :

- Manuel d'installation (cf. rapport M22 Annexe 1)
- Manuel d'utilisation (cf. rapport M22 Annexe 2)

1.3.9.5 Visualisation

La recherche et le jeu musical de sons à partir d'une grande base de données sonores est similaire à l'exploration de données en manipulant une representation graphique. Cette tache est bien décrite et beaucoup de recherches sur elle ont été fait dans le domaine de la visualisation d'information (Information Visualisation) (Tukey 1977). Shneiderman&Plaisant (2005) définissent le *Mantra de la visualisation d'informations* ainsi:

Overview, zoom and filter then details on demand

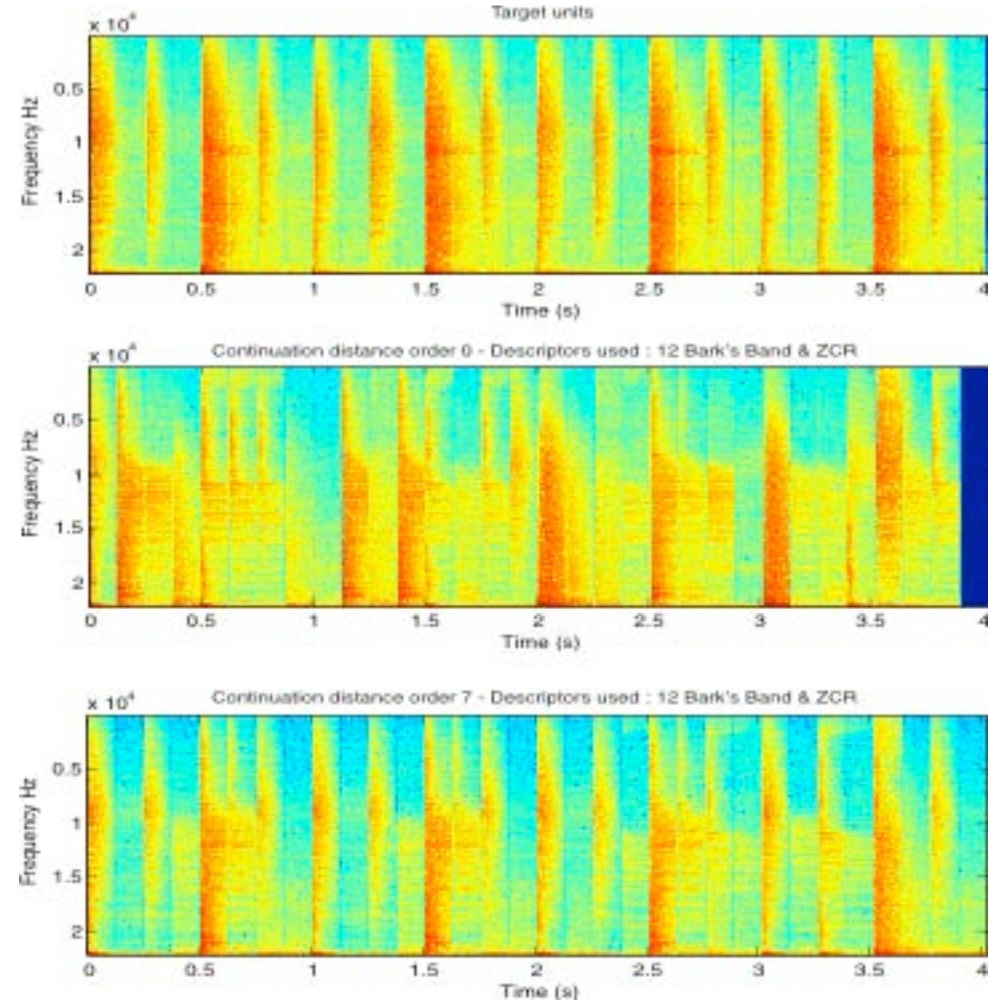
Pour permettre une exploration efficace de l'espace de sons défini par les descripteurs sonores, le système de visualisation de CataRT à été remodelé entièrement. Cette amélioration de l'architecture du modèle graphique est basée sur trois espaces de coordonnés (model, world, device coordinate spaces) et des transformations affines ou mapping non-linéaires entre espaces. Cette architecture permet une intégration facile de fonctionnalités de *zoom & pan* et un nouveau mode *tile* de mosaiques formés par la subdivision de l'affichage selon un ou deux descripteurs catégoriels.

1.3.9.6 Modélisation de la continuation de boucles rythmiques

Des recherches sur la modélisation du comportement temporel d'un corpus de sons et de ses transitions a été effectué [Schwarz08a], basées sur une distance dite de continuation à partir du corpus. Cette distance exprime les comportements temporels inhérents au corpus et peut être appliquée pour guider la synthèse par navigation dans un espace de descripteurs tel qu'offre CataRT (cf. SP3-4 – Variation interactive de boucles rythmiques) : ces recherches ont été effectués au sein de l'application CataRT et ont résulté dans des parties de la librairie *DistLib* et des nouvelles fonctionnalités de CataRT.

Une évaluation de l'efficacité de l'effet de continuation a été effectuée pour estimer l'influence des paramètres. Sur un corpus de 78 boucles de batterie, une boucle cible a été resynthétisée avec une adaptation des paramètres manuelle. Par rapport à la synthèse sans

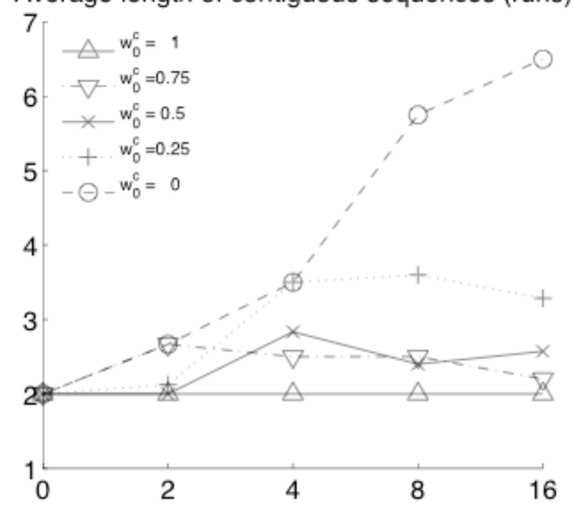
continuation, le résultat perçu montrait une plus grande régularité du résultat, comme on peut le voir dans ces spectrogrammes :

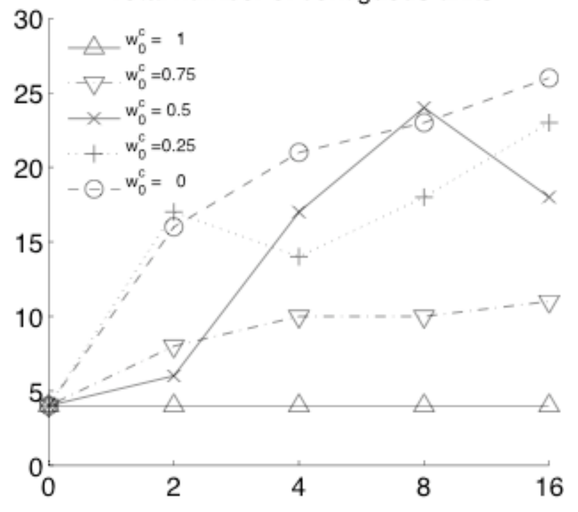


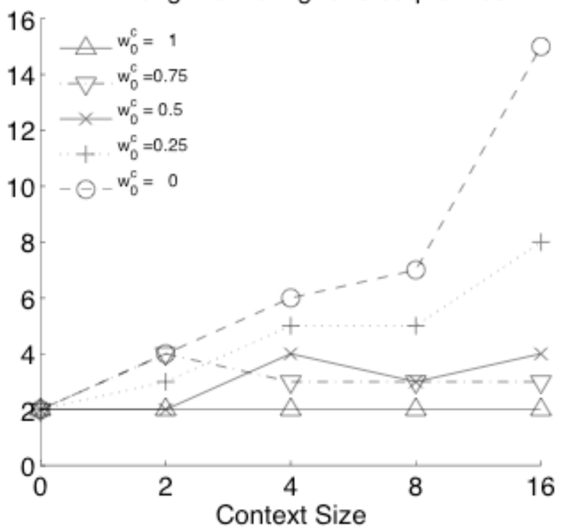
Influence du contexte sur la sélection d'une boucle de batterie. En haut : cible, milieu: sélection sans contexte, en bas: sélection avec contexte.

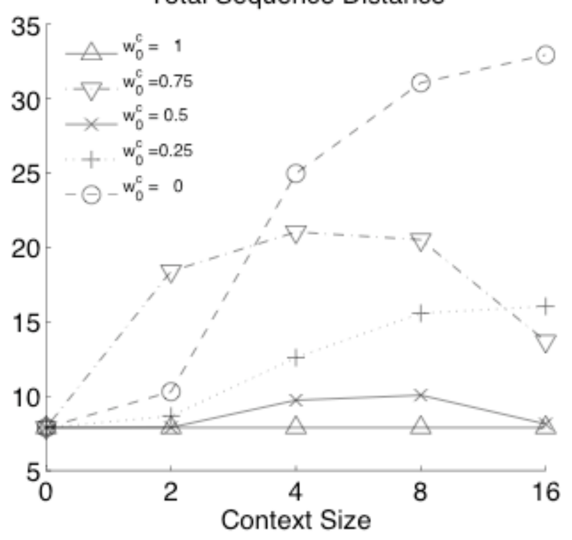
Puis, pour estimer l'influence des paramètres, des mesures objectives ont été prises par rapport au nombre et à la longueur des séquences pris en contiguïté dans le corpus. Ces résultats sont montrés dans la figure suivante :


Average length of contiguous sequences (runs)


$w_0^c = 1$
$w_0^c = 0.75$
$w_0^c = 0.5$
$w_0^c = 0.25$
$w_0^c = 0$
1
2
3
4
5
6
7
0
2
4
8
16


Total number of contiguous units
$w_0^c = 1$
$w_0^c = 0.75$
$w_0^c = 0.5$
$w_0^c = 0.25$
$w_0^c = 0$
0
5
10
15
20
25
30
0
2
4
8
16


Max length of contiguous sequences
$w_0^c = 1$
$w_0^c = 0.75$
$w_0^c = 0.5$
$w_0^c = 0.25$
$w_0^c = 0$
0
2
4
6
8
10
12
14
16
0
2
4
8
16
Context Size


Total Sequence Distance
$w_0^c = 1$
$w_0^c = 0.75$
$w_0^c = 0.5$
$w_0^c = 0.25$
$w_0^c = 0$
5
10
15
20
25
30
35
0
2
4
8
16
Context Size

1.4 Analyse et traitement audio

1.4.1 SP3-1 – Moteur d'analyse et de transformations sonores temps réel

Responsable : Ircam-AnaSyn

L'objectif des travaux présentés dans cette section est le développement d'une bibliothèque d'analyse et traitement du son pour toutes les plateformes ciblées, plus précisément pour Windows et MacOSX. En partant de la bibliothèque SuperVP de l'IRCAM qui permet les transformations et analyses des sons nous avions proposé plusieurs sujets visant à améliorer ou étendre les fonctions existantes. Dans la suite, nous discutons les différents points qui ont été étudiés et commentons les résultats obtenus.

1.4.1.2 Module de base

La livraison des bibliothèques existantes avec les fonctionnalités de départ a été souhaitée par le partenaire Univers Sons afin de pouvoir faire des tests initiaux des fonctionnalités existantes. Ce en vue de permettre de localiser rapidement les problèmes fondamentaux du design de la bibliothèque et aussi faciliter les tests des nouvelles structures qui sont à établir pour l'application finale.

La création de la bibliothèque pour le système MacOSX a été effectuée sans problème et nous avons pu délivrer la bibliothèque dynamique, comme prévu, fin juin 2007. Concernant la bibliothèque pour le système Windows nous avions rencontré plusieurs problèmes qui ont significativement retardé la livraison d'une version Windows du module de base. Les problèmes étaient liés à :

- Un retard de la livraison du logiciel nécessaire pour installation de Windows sur les ordinateurs de l'IRCAM,
- Des problèmes de performance sous Windows dû au manque des bibliothèques vectorisées (FFT, et fonctions mathématiques sin/cos/log/exp) ,
- Des problèmes d'optimisation du compilateur MSVC.

Afin d'améliorer la performance des algorithmes sous Windows, nous avons alors développé un certain nombre de fonctions et extensions des bibliothèques existant dans SuperVP afin de pouvoir obtenir une efficacité comparable a la version Mac :

- Une couche de compatibilité unifiant l'utilisation de la bibliothèque *pthreads* sous Mac OS et des threads natifs sous Windows,
- Une extension de l'interface de la bibliothèque *libFFT* utilisé dans SuperVP afin d'effectuer le calcul de la transformation Fourier rapide. Après l'extension de la bibliothèque *libFFT* peut être configuré à utiliser des bibliothèques MKL d'Intel, ACML de AMD et FFTW3,
- Les fonctions mathématiques (*sin*, *cos*, *log* et *exp*) utilisant les commandes SIMD avec un interface compatible avec les fonctions de la *vecLib* qui sont utilisées sous Mac OS X,

À cause des nombreuses bogues rencontrées avec le compilateur MSVC++, notamment l'impossibilité d'activer les options « fast math » et « SSE » ensemble, nous avons acheté le compilateur C++ d'Intel pour les compilations sous Windows.

1.4.1.3 Transposition en domaine spectral

L'implémentation de la transposition en domaine fréquentiel a été une tâche beaucoup plus lourde que prévu. Une première implémentation qui a été basée sur les propositions de Dolson et Laroche [Laroche/Dolson:99], a été finalisée en M10. Cette implémentation n'a pas

encore donné une qualité de transformation suffisante. A l'échéance M18 et en intégrant les modes de transformation dédiés à la préservation de la forme d'onde et des transitoires nous avons pu obtenir une qualité très proche de l'algorithme traditionnel. Malheureusement, comme nous avons expliqué dans le rapport précédant, le nouvel algorithme est moins robuste par rapport à la distance des trames de synthèse qui ne doit pas dépasser une sixième de la taille de la fenêtre (contre une troisième de la taille de la fenêtre pour l'algorithme traditionnel). Ce problème est dû au fait que le déplacement des pics change le rangement entre les bins dans les trames voisines de sorte que la base du calcul des phases modifiées qui est utilisée dans le vocodeur de phase est altéré.

Nous avons alors cherché à renforcer la cohérence de l'algorithme en estimant les paramètres sinusoïdaux pour chaque pic spectral, et à ensuite les modifier et à resynthétiser les sinusoïdes modifiées en utilisant la synthèse des sinusoïdes en domaine spectral [Rodet92b]. Les paramètres des pics qui sont utilisés sont les paramètres sinusoïdaux fondamentaux (amplitude, fréquence, phase) et en plus la pente de la fréquence. La resynthèse se fait à partir d'un tableau de pics spectraux lequel est précalculé pour une grille de pentes et ensuite interpolé de façon linéaire [Roebel08a]. Cela nous permet de prendre en compte le changement de la pente fréquentielle due à la transposition. L'algorithme d'analyses sinusoïdales et de la resynthèse à partir du tableau a été mis en place en août 2008. Une nouvelle version de la transposition par analyse/resynthèse sinusoïdale a été mis en place en septembre et octobre 2008. Ce nouvel algorithme n'intégrait pas encore les fonctionnalités avancées (comme la préservation des attaques et de la forme d'onde). Par contre, il nous a permis de constater que l'algorithme travaillant avec les paramètres sinusoïdaux est aussi efficace que la version basée sur le déplacement des pics spectraux. Un avantage de la nouvelle approche est le fait que la forme du pic est changée de façon cohérente avec la transposition (changement de la pente de la fréquence). Les travaux ont été poursuivis et les modes de transformations liées à la préservation des transitoires et à la préservation de la forme d'onde ont été intégrés avec la transposition utilisant les modèles sinusoïdaux. L'utilisation des paramètres sinusoïdaux nous a permis des réductions du temps de calcul pour le mode préservation de la forme d'onde. Ces travaux ont été finalisés en mai 2009.

[Laroche/Dolson:99] Laroche, J., Dolson, M., « New phase-vocoder techniques for real-time pitch shifting, chorusing, harmonizing and other exotic audio modifications », Journal of the AES, 47:11, pp. 928-936, 1999.

[Rodet92b] Freed, A., Rodet, X., Depalle, P., « Synthesis and Control of Hundreds of Sinusoidal Partials on a Desktop Computer without Custom Hardware », International Conference on Signal Processing Applications & Technology (ICSPAT) 1992.

[Roebel08a] Roebel, A. « Frequency-Slope Estimation and Its Application to Parameter Estimation for Non-Stationary Sinusoids », Computer Music Journal, 32:2, pp 68-79, 2008.

1.4.1.4 Traitement avec une taille de fenêtre variable

La taille de la fenêtre d'analyse est le paramètre fondamental qui détermine la qualité obtenue pour toutes les analyses et/ou traitements spectraux. Dans la majorité des cas, les algorithmes pour la transformation de sons travaillent aujourd'hui avec une résolution temps/fréquence fixe et donnée par l'utilisateur. La résolution peut varier en fonction de la fréquence (ondelettes) mais elle ne peut pas varier au cours du temps pour s'adapter aux propriétés locales du signal. Dans le cadre de ce sous-projet, nous avions proposé d'étudier les possibilités de travailler avec une résolution variable adaptée aux propriétés du son. Pour ce faire, il est nécessaire à distinguer deux situations différentes : le cas des sons monophoniques et le cas des sons polyphoniques.

En rapport avec les sons polyphoniques nous avons étudié plusieurs algorithmes proposés dans la littérature pour l'évaluation de la taille de la fenêtre à partir du spectrogramme. Nous avons comparé la mesure d'entropie généralisé de Renyi [Baraniuk/Flandrin/Janssen/Michel:01], la mesure proposée dans [Lukin/Todd:06], et

également la mesure proposée dans [Jones/Baraniuk:94]. L'évaluation expérimentale a été effectuée en utilisant plusieurs signaux synthétiques qui représentaient les caractéristiques des signaux monophoniques dynamiques. Les expériences ont montré que :

- Les différentes mesures donnent des résultats similaires. Par contre les résultats dépendent très fortement du bruit contenu dans le signal.
- Les résultats ne sont pas toujours cohérents avec les fenêtres considérées comme optimales,
- Les résultats sont plutôt adaptés pour la visualisation des spectrogrammes ; pour le traitement et en utilisant des techniques comme la préservation des transitoires, les résultats ne sont pas directement applicables.

Nous pouvons en conclure que plus d'expérimentation sera nécessaire avant que ces mesures puissent êtres utilisés dans des applications réelles. Les résultats sont décrits plus exactement dans le rapport de stage de [Boudard07a].

Pour le cas des signaux monophoniques, la détermination de la taille de fenêtre peut se faire directement à partir de la fréquence fondamentale. Pour les régions non-voisées, la taille de la fenêtre a moins d'importance et elle peut être choisie par interpolation entre les valeurs adaptées aux régions voisées voisinées. Afin de pouvoir utiliser une taille de fenêtre adaptée localement à la fréquence fondamentale nous avons modifié tous les algorithmes pour qu'ils puissent travailler avec des fenêtres variables.

L'évaluation expérimentale montre que les avantages de la taille de la fenêtre variable sont intéressants surtout pour les sons de parole qui contiennent souvent des fins de mots avec une fréquence fondamentale basse.

À cause des difficultés rencontrées en rapport avec les travaux sur la transposition en domaine fréquentiel, nous avons dû décaler les travaux concernant la modification dynamique de la taille de la fenêtre. Ces travaux ont été finalisés en février 2008. Les travaux concernant l'adaptation automatique de la taille de la fenêtre d'analyse seront poursuivis dans le cadre d'une thèse.

[Baraniuk/Flandrin/Janssen/Michel:01] Baraniuk, R.G., Flandrin, P., Janssen, A., Michel, O., « Measuring Time-Frequency Information Content Using the Rényi Entropies», IEEE Trans. on Information Tech. 47(4) :1391-1410, 2001.

[Lukin/Todd:06] Lukin, A. Todd, J., « Adaptative Time-Frequency Resolution for Analysis and Processing of Audio », AES Convention paper 6717, 2006.

[Jones/Baraniuk:94] Jones, D.L., Baraniuk, R.G., « A simple scheme for adapting time frequency representations », IEEE Trans on Signal Processing, 42:12, pp. 3530—3535, 1994.

[Boudard07a] Boudard, L., « Adaptation des tailles de fenêtre pour les représentations temps frequences», rapport de stage Ircam, 2007.

1.4.1.5 Amélioration/qualification du F0 pour le contrôle

<u>Extraction de la fréquence fondamentale</u>

La bibliothèque pour l'estimation de la fréquence fondamentale utilisée dans le moteur de calcul SuperVP a été évaluée à plusieurs reprises et les paramètres ont été adaptés de sorte que pour le cas général où la source sonore (l'instrument) n'est pas connue, nous obtenons les meilleurs résultats. Puisque les instruments de musique ont des caractéristiques spectrales assez variables, il est évident que la connaissance de l'instrument nous permettrait de choisir des paramètres spécialement adaptés. Puisque l'algorithme de l'estimation F0 utilisé est basé sur une comparaison pondérée de plusieurs critères d'évaluation des F0 hypothétiques, il est facilement imaginable de créer des paramètres adaptés aux instruments. Pour réaliser ce projet nous avons créé une petite collection de signaux annotés pour les instruments critiques (flûte, piccolo, basson, hautbois, clarinette) et nous avons optimisé les paramètres de notre algorithme pour ces classes d'instruments. La comparaison des résultats obtenus avec les paramètres par défaut et les paramètres adaptés par instruments montre une réduction d'erreurs significative. Avec les paramètres

adaptés, les erreurs systématiques qui étaient observées pour plusieurs notes de chacun de ces instruments ont disparu. Les erreurs restantes sont toutes liées à des conditions très difficiles comme les attaques ou les fins des notes ou l'énergie des composants harmoniques est trop faible.

<u>Qualification de la fréquence fondamentale</u>

Afin de pouvoir utiliser la fréquence fondamentale estimée comme paramètre de contrôle, nous avions proposé d'améliorer l'algorithme existant en y ajoutant des fonctionnalités qui permettent la qualification de l'estimation. Pour cela nous avons choisi une méthode très proche de la méthode de l'algorithme yin [Cheveigne/Kawahara, 2002] qui donne des bons résultats pour l'évaluation de la périodicité du signal en se basant sur la fonction d'auto corrélation (*ACF*) du signal à la position de la période estimée. Afin de normaliser les résultats de l'ACF, nous utilisons une taille de fenêtre adaptée à la période estimée pour le calcul de l'ACF. Le nombre des périodes utilisées pour le calcul de l'ACF peut être choisi par l'utilisateur. Un deuxième problème qui peut affecter les résultats de l'ACF est la variation fréquentielle. Plus la fréquence fondamentale varie, plus l'ACF va être basse, même si le signal est très harmonique. Le choix d'une taille de fenêtre adaptée à la fondamentale améliore la situation. Néanmoins, dans notre algorithme nous avons intégré une démodulation de la fréquence fondamentale estimée si sa variation sur le segment utilisé pour la qualification de la périodicité est trop importante. Puisque cette démodulation est relativement coûteuse, elle est optionnelle. Un deuxième problème avec l'utilisation de l'ACF est le cas du bruit en fréquence basse. Si le signal contient une composante dominante à une fréquence en dessous de la fréquence fondamentale détectée, l'ACF n'est plus fiable. Nous avons alors modifié le critère en ajoutant un facteur qui tient compte de cette situation.

Après avoir implémenté l'algorithme de la qualification de la f0, nous l'avons comparé de façon qualitative avec les résultats obtenus avec d'autres algorithmes qui étaient à notre disposition. Les résultats peuvent être résumés de la manière suivante : la nouvelle version de la qualification de la périodicité est très robuste, elle permet la sélection automatique des segments voisés par un seuillage simple sur la valeur de la confiance de périodicité qui est calculée. En comparaison avec la valeur de qualification existant précédemment dans le moteur de calcul SuperVP, nous avons significativement amélioré les résultats En comparaison avec les mesures de qualification existant dans l'algorithme yin, nos résultats semblent légèrement plus robustes du fait que la taille du segment utilisé pour le calcul de l'acf est plus grande et que nous traitons correctement la situation avec composantes avec une fréquence basse dominante.

La création des préréglages a été effectuée en mai et juin 2008. Les travaux d'implémentation de la qualification de la périodicité ont été entrepris à partir de novembre 2007 et ont été finalisés en mars 2008.

[Cheveigne/Kawahara, 2002] Cheveigne, A., Kawahara, H. « YIN, a fundamental frequency estimator for speech and music », Journal of the Acoustical Society of America, 111:4, pp. 1917-1930, 2002.

1.4.1.6 Réorganisation des modules d'analyse

Le module d'analyse centrale de la bibliothèque SuperVP ne contient qu'un seul bloc de sorte qu'il n'est pas possible de séparer une analyse en plusieurs pas. Par conséquent, si plusieurs analyses doivent être faites pour le même son et aucun partage du calcul n'est possible. Afin de pouvoir extraire le calcul de la transformée de Fourier, nous avons réorganisé l'implémentation du module d'analyse. Le module a été séparé en deux. Ces deux modules sont normalement enchaînés. Ces travaux ont été entrepris en décembre 2007.

Sur la base de cette nouvelle organisation, nous avons pu créer l'interface analyses multiples pour la bibliothèque SuperVP qui permet alors les analyses : DFT, enveloppe spectrale, fréquence fondamentale, fréquence de coupure voisée/non voisée et détection des marques des débuts de notes sans avoir a dupliquer les DFT.

Les avantages de la nouvelle organisation interne ont été déjà valorisés dans le cadre d'un projet industriel de nouveau avec la société *MakeMusic* qui nous a sollicités pour créer une bibliothèque de nettoyage des sons d'accompagnements afin de pourvoir réduire les erreurs de l'estimation de la fréquence fondamentale dans leur produit de formation pour les instrumentistes. Grâce à la réorganisation effectuée, nous avons pu créer une bibliothèque qui effectuait d'abord la transformée de Fourier puis insérait un nouveau plug-in pour la suppression des sons d'accompagnements et après effectuait soit une analyse de la fréquence fondamentale soit une resynthèse du signal nettoyé.

Suivant les demandes fortes du partenaire Univers Sons qui a considéré que la réduction du temps de calcul serait l'un des objectifs prioritaires pour l'utilisation du moteur dans leurs produits, nous nous sommes concentrés sur la finalisation d'un module de synthèse à partir de pré-analyses qui semblait avoir le plus grandes potentiel pour la réduction du temps de calcul. Ce module de synthèse permet aussi de synthétiser des trames pré-analysées en les enchaînant de façon libre. Cela permettra un nouveau mode de synthèse granulaire en utilisant la superposition par vocodeur de phase pour un rendement lisse des grains voisins.

Ces travaux ont abouti fin juin 2009 à la réalisation d'une nouvelle interface pour la synthèse qui permet la synthèse à partir des analyses et ce en donnant accès à tous les contrôles et algorithmes existant dans les objets standard.

La réorganisation du code a été effectuée en décembre 2007. La création de l'interface analyses multiples a été effectuée en août et septembre 2008. Ce fonctionnalité utilise pour le projet VoiceForger, une bibliothèque pour la transformation de la voix qui est développée dans le projet ANR « Affective Avatar ». La nouvelle interface pour la synthèse à partir d'analyse a été finalisée en juin 2009.

1.4.1.7 Amélioration du remixage des composantes

Le remixage des composantes sinusoïdales et du bruit nécessite une classification des pics spectraux qui est suivi par une multiplication des pics de la classe à remixer par le facteur de remixage. Le paramètre actuellement accessible dans la bibliothèque qui sert à modifier le résultat de la classification est l'erreur de mauvaise classification des pics bruités comme sinusoïdes. Dans nos études précédentes, nous avons distingué 2 caractéristiques des pics spectraux qui sont très efficaces pour distinguer les deux classes des pics spectraux, d'une part les pics générés par une sinusoïde seule avec variation de la fréquence et de l'amplitude limitée et d'autre part les pics spectraux générés par le bruit. Les deux descripteurs sont la largeur de la bande du pic normalisé (NBD) et la durée du signal liée au pic (DD).

Nous avons été invités à soumettre un article sur ce sujet au Computer Music Journal et alors nous avons réorganisé notre planning et ont investi une part du temps consacré au remixage dans l'écriture de l'article [Zivanovic08a].

Ensuite nous avons abordé la question du mécanisme de la classification basé sur le descripteur NBD. Ayant établi expérimentalement le fait que le descripteur NBD est le meilleur descripteur pour la distinction entre pics sinusoïdaux et pics bruités, il nous semblait important à expliquer si le fonctionnement du descripteur peut être compris dans le domaine temporel.

L'étude théorique de cette question a été entreprise en collaboration avec M. Zivanovic de l'Université de Navarra en Espagne. Cette étude nous a montré que le descripteur NBD mesure la durée de l'enveloppe du signal liée aux pics. Grâce à la normalisation par la largeur de la bande du pic, la durée n'est pas mesurée en temps mais en échantillons avec un taux d'échantillonnage adapté à la largeur de bande du pic observé. La compréhension de ce mécanisme nous a donné des nouvelles idées pour le contrôle des seuils de la classification car il nous semble que nous pouvons transformer les deux seuils liés avec la durée temporelle du signal dans un seul seuil qui limite la différence absolue entre la durée de la sinusoïde stationnaire fenêtrée et la durée du signal liée aux pics. En raison de la

préparation de l'article et des travaux nécessaires à l'amélioration de la transposition en domaine fréquentiel, nous avons dû supprimer l'implémentation et l'évaluation de cette nouvelle formulation du classificateur.

La réorganisation des fonctions nécessaires pour la transformation dans le domaine spectral nous a amené à implémenter des fonctions d'analyse/synthèse sinusoïdale en domaine spectrale dans le vocodeur de phase (voir SP3-1-2). Ces nouvelles fonctions nous ont permis de réaliser une amélioration significative du module de remixage. Nous y avons ajouté un mode remixage avancé dans lequel pour chaque pic sinusoïdal les paramètres sinusoïdaux sont estimés comme le remixage des sinusoïdes ne s'applique que a la partie sinusoïdale du pic. Contrairement au fonctionnement auparavant ce nouveau mode réduira l'effet de la taille de la fenêtre au résiduel et approchera les résultats obtenus avec le module de remixage du vocodeur de phase aux résultats obtenus avec un vrai modèle sinusoïdal.

Les travaux concernant la publication ont été effectués en novembre et décembre 2007, les travaux théoriques concernant le fonctionnement de la classification ont été conduits en printemps 2008 et l'implémentation du module de rémixage fondé sur le modèle sinusoïdal a été effectué en août 2008.

[Zivanovic08a] Zivanovic, M., Röbel, A., Rodet X., « Adaptive Threshold Determination for Spectral Peak Classification ». Computer Music Journal. 32:2, pp. 57-67, 2008.

1.4.1.8 Détection, préservation et synthèse de modulations

À cause des problèmes rencontrés dans le cadre de la transposition en domaine fréquentiel, les travaux prévus pour la détection et modification des modulations ont été décalés d'à peu près 2 mois. Le problème du traitement des modulations est peu discuté dans la littérature. Nous avons alors commencé en identifiant les différentes tâches liées au problème. Nous distinguons :

- La modification des paramètres de la modulation dans le signal,
- L'adaptation des modifications demandées par l'utilisateur au contexte musical,
- La détection des segments modulés qui sont perçus comme unités musicales.

Nous avons commencé l'étude des transformations des modulations en supposant que nous connaissons les parties modulées du signal et que les changements sont cohérents avec le contexte musical, tel qu'ils peuvent être effectué sans modification. Alors nous nous sommes concentré sur le sujet (1) de la liste, la transformation des modulations de façon indépendante du signal.

Deux transformations des modulations ont été sélectionnées pour une étude précise. La première est liée au changement de la fréquence de modulation de façon indépendante de la durée du son. Dans cette direction, nous espérons pouvoir établir une méthode qui servira comme moyen pour la modification de la fréquence des modulations pour les segments sélectionnés sans avoir un impact sur la durée de ces segments. La deuxième est liée à l'amplitude de la modulation qui servira à pouvoir augmenter, réduire ou annulé les modulations. Ces deux méthodes sont d'abord utilisées pour le cas du vibrato qui nous semble plus important et plus difficile.

Le premier pas pour les deux modifications du vibrato est la séparation du geste de base de la fréquence fondamentale (mélodie) des changements ornementaux (modulations). Puisque le vibrato est considéré comme un effet périodique, nous supposons que cet effet peut être caractérisé par une fréquence fondamentale des changements de la fréquence fondamentale du signal. Nous appelons cette fréquence fondamentale du changement de la fréquence fondamentale le $F0_f$. Il est connu que la valeur de $F0_f$ est normalement située autour de 6Hz. En raison du fait que cette modulation est quasi périodique, nous attendons un certain nombre des partiels en relation fréquentielle à peu près harmonique avec $F0_f$.

Pour accomplir la séparation entre mélodie et modulation, nous avons alors développé une méthode de filtrage basée sur les modèles sinusoïdaux qui nous permettent de supprimer la

mélodie de l'évolution de la fréquence fondamentale en supposant que les segments traités ne contiennent pas de changements de notes.

Sur la base de la séparation obtenue, nous avons pu mettre en œuvre une méthode de modification indépendante de la durée et de la fréquence du vibrato. Pour cela nous utilisons le vocodeur de phase pour sauter et répéter les périodes du vibrato et en même temps pour effectuer une dilatation ou compression. Cette méthode donne des résultats très convaincants, elle nécessite par contre que les points de répétition soient sélectionnés automatiquement. Pour cela il convient d'utiliser les points de passage par 0 de la courbe de la fréquence fondamentale contenant seulement la partie de modulation.

Concernant le changement de l'amplitude de la modulation nous avons étudié une approche utilisant une transposition dynamique. Les résultats ont montré que même si la transposition dynamique enlève la modulation fréquentielle de façon efficace les résultats restent peu convaincants car les modifications de l'enveloppe et la modulation de l'amplitude induites par le vibrato restent perceptiblement très présentes. Nous avons identifié deux sources possible pour ces modulations induites : la première est dû au résonateur de l'instrument qui est supposé fixe et qui va alors moduler les partiels qui ont été déplacés en fréquence. La deuxième source est liée au fait que le résonateur lui-même change avec les modifications de la fréquence fondamentale.

Nous avons étudié la performance d'un des produits du marché (*Melodyne*) et nous avons trouvé que ce logiciel n'arrive pas à supprimer le trémolo induit. Nous avons alors poursuivi en mettant en place deux approches pour remédier à ce problème. Le premier consiste à préserver l'enveloppe spectrale lors de la transposition. Cela devrait enlever la modulation d'amplitude induit si l'enveloppe spectrale ne change pas avec la modulation. Les résultats ont montré que cette approche était peu effective, et cela nous conduit à la conclusion que l'amplitude et l'enveloppe spectrale changent avec la fréquence fondamentale.

Nous montrons un exemple de ce résultat en Figure 28. Dans ce figure on voit le spectrogramme et l'enveloppe spectrale d'un son de flûte avec vibrato. Le vibrato est facile à repérer dans le spectrogramme et de même dans l'évolution temporelle de l'enveloppe spectrale.

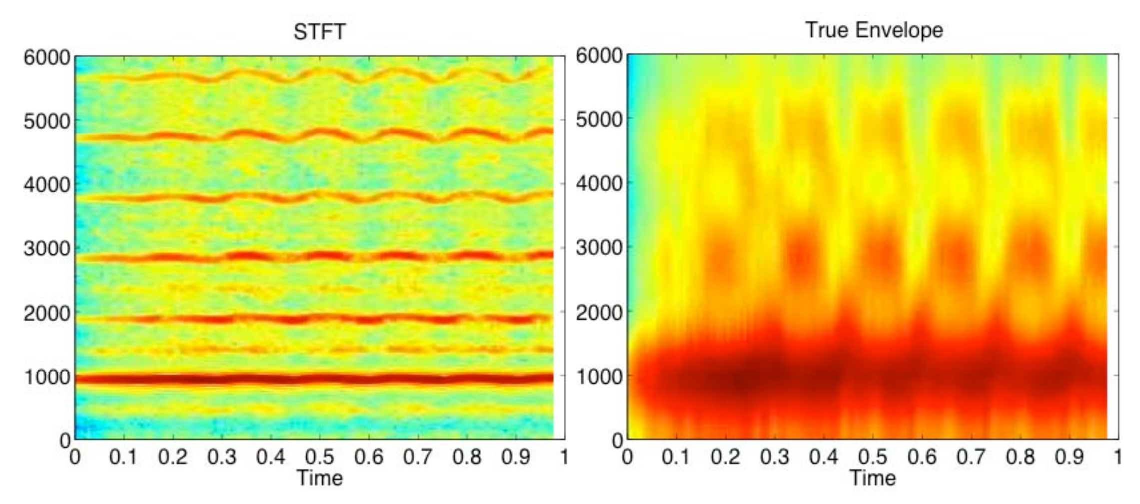


Figure 28 - Spectrogramme (gauche) et enveloppe spectral (droite) d'un son d'un flute avec vibrato

En Figure 29, on voit les mêmes analyses pour ce son après transposition dynamique qui est adaptée de manière que la modulation de la fréquence fondamentale est enlevée. Pour ce résultat, nous avons choisi à préserver l'enveloppe spectrale. Le vibrato est réduit de façon significative, mais la modulation de l'amplitude des partiels et de l'enveloppe spectrale reste importante.

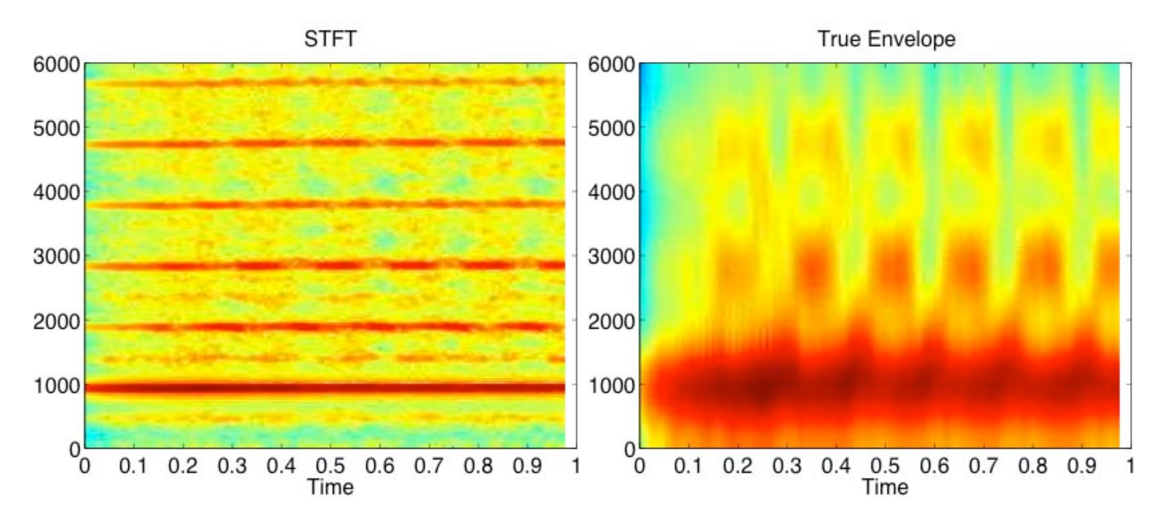


Figure 29 - Spectrogramme (gauche) et enveloppe spectral (droite) d'un son d'un flûte après transposition dynamique enlevant le vibrato. L'enveloppe spectrale est préservé.

Afin de pouvoir réduire cette modulation restante, nous avons développé plusieurs procédures de filtrage adaptatif, qui compensent le changement de l'enveloppe spectrale qui est induit par la modulation de la fréquence fondamentale. Le spectre et l'enveloppe du son de la flûte est montre en Figure 30. Les modulations d'amplitude des partiels et de l'enveloppe sont significativement réduites.

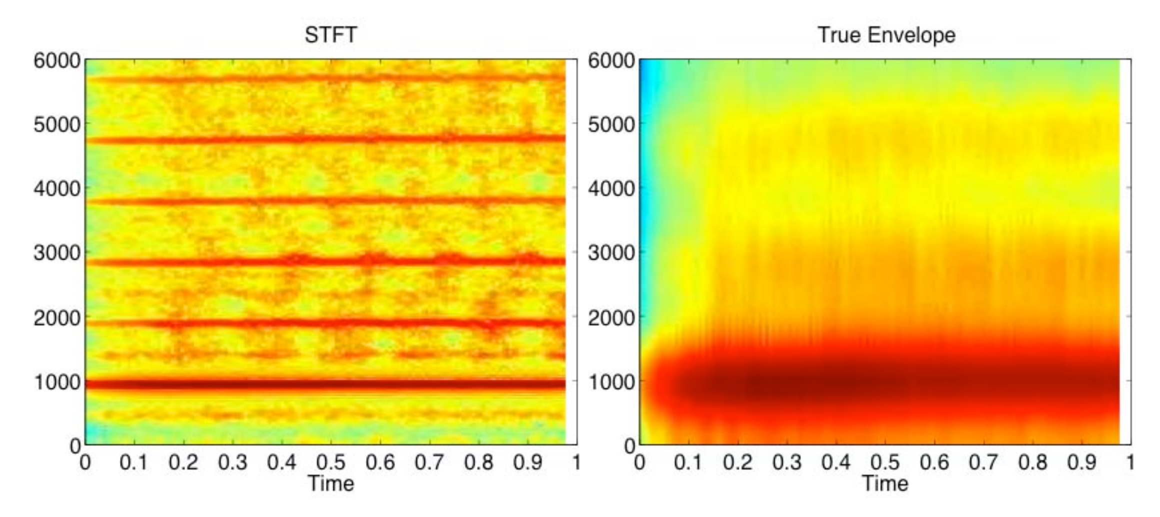


Figure 30 - Spectrogramme (gauche) et enveloppe spectral (droite) d'un son d'un flûte après transposition dynamique avec préservation de l'enveloppe spectrale et filtre adaptive.

Cette dernière approche donne des résultats très satisfaisants si le niveau de bruit du son est faible. Les petites modulations restantes sont dues aux erreurs d'estimation de la fréquence fondamentale. Pour les instruments avec une forte composante bruit une modulation d'amplitude dans la composante bruit reste dans le son. Cela est perceptivement très gênant. Ces résultats nous ont amenés à implémenter une démodulation individuelle des composantes sinusoïdales et bruitées. Les résultats des travaux sur la modification des modulations sont actuellement implémentés en Matlab. Ils sont décrits plus précisément dans [Maller08]. Ces prototypes montrent les améliorations obtenues, par contre ils ne sont pas encore suffisamment robustes pour être utilisables dans un produit. Les travaux initiaux concernant la démodulation générale ont été effectués entre mars et juillet 2008. L'extension de la méthode considérant les composantes sinusoïdales et bruite de façon indépendante a été effectuée entre mars et juin 2009.

[Maller08a] Maller, S., « Transformation des sons musicaux avec préservation des articulations musicales », rapport de stage Ircam, 2008.

1.4.1.9 Traitement des régions bruitées

À cause des multiples problèmes en relation avec l'implémentation de la transposition en domaine fréquentiel et en accord avec le partenaire Univers Sons nous avons décidé qu'une continuation et finalisation des travaux concernant la transposition serait plus intéressante que les travaux envisagés dans ce sous projet. Nous avons alors réaffecté le temps correspondant aux autres travaux.

1.4.2 SP3-2 – Moteur de synthèse granulaire temps réel

Responsable : Ircam-IMTR

Délivrables

- D-SP3-2-1 : Prototypes de différents modes de synthèse granulaire.
- D-SP3-2-2 : Première version du moteur
- D-SP3-2-3 : Module final

Le moteur de synthèse granulaire à été réalisé, sous forme d'une bibliothèque C sous le nom *ZsaZsa*.

Une première version de l'API de la bibliothèque a été livrée et discutée entre l'IRCAM et Univers Sons début 2008. Suite à ces échanges, une première implémentation du moteur en code source a été distribuée mi-avril 2008 fournissant un seul mode de fonctionnement pour la synthèse granulaire simple (*asynchrone*). La bibliothèque a été intégrée dans Max/MSP et dans une application prototype réalisé par Univers Sons.

Dans la version diffusée en octobre 2008, le moteur intègre trois modes de fonctionnement :

- Synthèse granulaire *asynchrone*
- Synthèse granulaire *synchrone*
  (nécessitant des marqueurs de forme d'onde élémentaire)
- Synthèse *concaténative*
  (nécessitant des marqueurs de segments)

La synthèse en ces trois modes est contrôlée par un ensemble de paramètres communs et quelques paramètres propres au mode de synthèse granulaire synchrone :

- Paramètres de contrôle de la synthèse (valeurs moyennes instantanées et amplitudes de variations aléatoires pour chaque paramètre) :
  - Position dans le fichier source
  - Fréquence ou période de grains
  - Paramètres de durée de grains (avant et après la position)
  - Transposition ou re-échantillonnage de grains
  - Niveau sonore de grains
  - Paramètres de fenêtrage (type et durées du début et de la fin)
- Paramètres de contrôle de la lecture par un ensemble de paramètres
  - Position de début et de fin de lecture
  - Vitesse de lecture
  - Mode de lecture (lecture simple ou en boucle)
- Paramètres de contrôle pour la synthèse granulaire synchrone
  - Variation aléatoire de la phase dans les segments non-voisés
  - Amplification/atténuation dans les segments non-voisés
  - Recouvrement (« overlap ») des grains dans les segments non-voisés

Au niveau le plus bas de l'implémentation le moteur re-synthétise des grains avec un schéma simple de paramétrage et de fenêtrage par sa position de début dans le fichier source (« onset time »), son durée (« duration »), et les durées d'attaque (« attack ») et de déclin (« release ») :

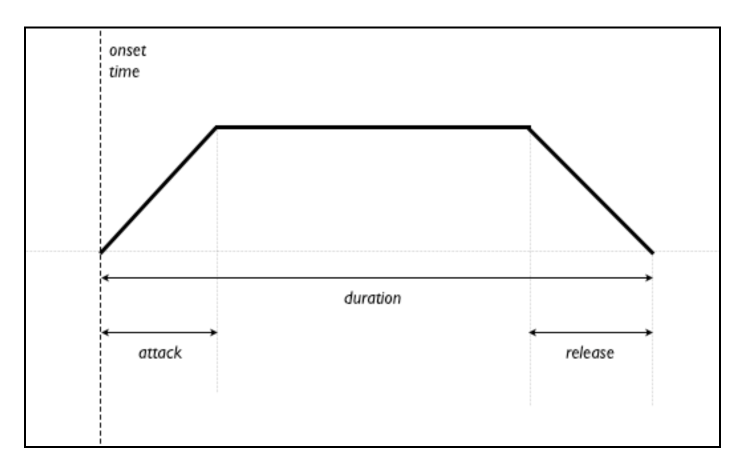


Figure 31 - Paramétrage de grains de bas niveau.

Pour pouvoir synchroniser précisément les grains, dans d'autres modes que la synthèse granulaire simple, dans l'interface de la bibliothèque, la durée du grain est partagée en deux parties (« head » et « tail ») avant et après la position de référence du grain dans le fichier source :

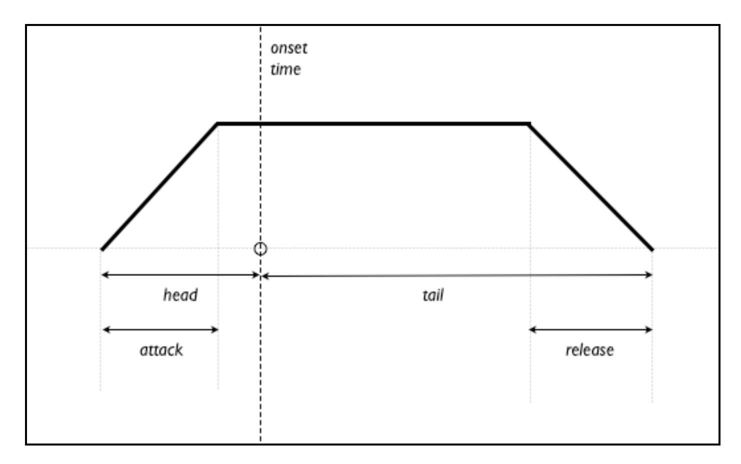


Figure 32 - Paramétrage de grains avec point de synchronisation (« head »).

De manière générale, la synthèse d'un grain par le moteur (sauf en mode granulaire simple) utilise trois marqueurs de position pour le temps de référence (« onset time ») ainsi que le début (« begin ») et la fin (« end ») de grain. En plus, le grain est déterminé par les quatre valeur de durée et de fenêtrage (« head », « tail », « attack » et « release », données en valeur absolue ou relative) :

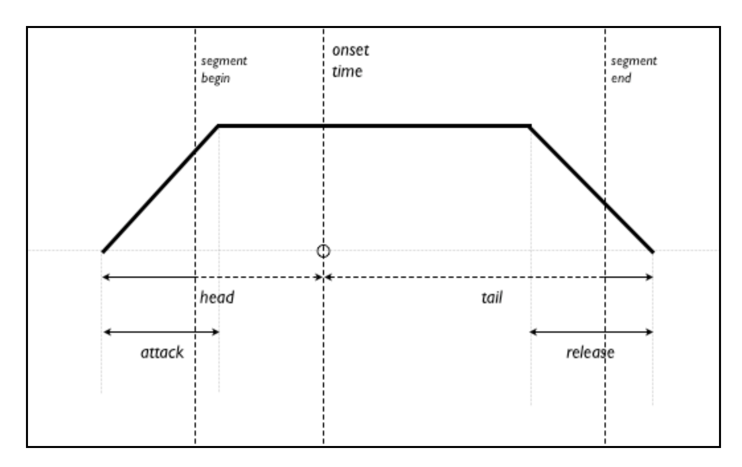


Figure 33 - Segmentation, paramétrage et fenêtrage génériques des grains synthétisés.

Pour des modes de synthèse spécifiques les marqueurs peuvent coïncider pour un grain donné (e.g. le temps de référence correspond au début) ainsi que pour des grains successifs (e.g. le fin grain correspond au début du grain prochain) :

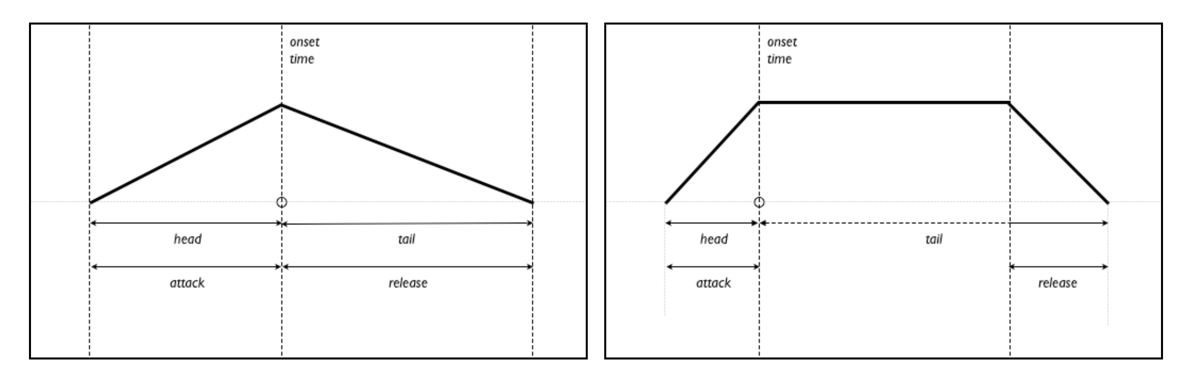


Figure 34 - Paramétrage et segmentation de grains dans le mode de synthèse granulaire synchrone et pour un cas spécifique de synthèse concaténative.

Le moteur implémente une interpolation optimisée (re-échantillonnage) afin d'obtenir une qualité maximale pour la transformation de la hauteur et du timbre. Cette interpolation permet en même temps d'ajuster la position de référence du grain et le temps (c'est à dire la période) de la synthèse avec une précision plus fine que le taux d'échantillonnage.

Le moteur a une architecture modulaire qui permet facilement diverses extensions futures et son intégration dans différents contextes. Le mode de synthèse par concaténation permet la définition d'une fonction de rappel (« callback », appelé pour chaque grain) pour faciliter l 'adaptation du moteur à des applications spécifiques comme la transformation de boucles rythmiques et la synthèse par corpus.

Dans l'interface de programmation du moteur, les marqueurs sont représentés en tant que vecteurs génériques de nombres réels pour permettre l'intégration facile du moteur dans différents contextes applicatifs (i.e. logiciels).

L'implémentation de la synthèse granulaire synchrone n'était pas prévue dans le projet initial, mais a été fortement souhaitée par Univers Sons et s'intègre de manière très homogène dans l'architecture du moteur.

Les tests conduits après une première intégration de la bibliothèque *ZsaZsa* dans l'échantillonneur, ont permis de valider ses fonctionnalités et d'établir une liste d'extensions souhaitées pour la finalisation. Ainsi, pendant la dernière période du projet, le moteur a pu être adapté au fonctionnement particulier de l'échantillonneur initialisant le moteur à chaque déclenchement de note. Cette finalisation concerne surtout l'optimisation de l'initialisation du moteur, la minimisation du retard entre le contrôle par l'échantillonneur et le son produit ainsi que l'option de traiter des son cycliques.

Ces extensions et optimisations du moteur ont été intégrées dans la version 2.1 et 2.2 de la bibliothèque *ZsaZsa* livrée aux partenaires du projet.

1.4.3 SP3-3 – Synthèse granulaire basée sur les descripteurs sonores

Responsable : Ircam-IMTR

Dans la première période du projet, un ensemble de fonctionnalités prototypes ont été présentées. Ils ont été réalisés avec le system CataRT basé sur la bibliothèque FTM & Co dans Max/MSP. Ces prototypes permettaient la synthèse granulaire d'un fichier sources en sélectionnant les grains par leur description selon un ensemble de descripteurs sonores extraits du son (à la place par simple position dans le fichier). Pour l'implémentation d'une application finale basée sur le moteur de synthèse granulaire, un ensemble réduit de fonctionnalités a été retenus. L'architecture mise en œuvre est décrite au 2.2.3 p. 95.

1.4.4 SP3-4 – Variation interactive de boucles rythmiques

Responsable : Ircam-IMTR

L'application finale de variation interactive de boucles rythmiques s'appuie sur les mêmes composants logiciels que la synthèse granulaire basée sur les descripteurs (voir SP3-3).

Concernant l'utilisation du moteur de synthèse granulaire, la différence entre ces deux applications consiste principalement en la segmentation du fichier source en pulsations rythmiques. L'application réalisée s'appuie sur des marqueurs de pulsation générés par le module logiciel *IRCAMbeat* (SP2-3), mais peut aussi importer des marqueurs générés par d'autres logiciels (e.g. *AudioSculpt*).

Au début du projet un prototype a été réalisé qui utilisait une modélisation temporelle des séquences rythmiques basée sur la définition d'une distance entre des séquences de descripteurs sonores [Cadars07][Schwarz08]. Cette modélisation n'a pas été retenue pour l'application prototype finale. Néanmoins, ces recherches représentent un résultat important du projet.

L'application réalisée intègre deux modes de fonctionnement :

- Recomposition temporelle des segments correspondant aux pulsations
- Remplacement des segments correspondant aux pulsations dans un fichier son donné par un autre basé sur la ressemblance (selon un ensemble de descripteurs choisi)

Le deuxième mode de fonctionnement correspond au « musaicing » prenant en compte une segmentation en pulsations rythmiques. L'unification des deux modes est possible si on considère la phase de pulsations (par rapport à la métrique ainsi que la structure rythmique à plus long terme) comme descripteurs.

[Cadars07] Cadars S., « Modélisation temporelle et synthèse concatenative de boucles rythmiques», *Rapport de Stage DEA ATIAM*, septembre 2007.

[Schwarz08] Schwarz D., Cadars S., Schnell N., « What Next? Continuation in Real-Time Corpus-Based Concatenative Synthesis », *Proceedings of the International Computer Music Conference (ICMC)*, Belfast, UK, août 2008.

1.4.5 SP3-5 – Transformations morphologiques interactives

Responsable : Ircam-IMTR

Après la première phase de prototypage du projet, les travaux dans le cadre des sous-projets SP3-1 à SP3-5 ont été focalisés sur le développement d'un ensemble cohérent de composants logiciels factorisant au maximum les fonctionnalités requises par les applications visés. En conséquence les différentes applications réalisées dans le cadre de ces sous-projets s'appuient sur les mêmes composants logiciels développés dans SP3 ainsi que des composants développés dans SP2.

1.4.5.2 Bibliothèque MuBu

L'outil qui a été était initialement planifié pour la *représentation et modification des descripteurs sonores et de la segmentation* uniquement dans le cadre de SP3-5, a été développé en tant que *conteneur de sons décris et annotés* sous forme d'une bibliothèque C++. La bibliothèque *MuBu* a finalement aussi servi pour la réalisation des prototypes et des applications dans les sous-projets SP3-2, SP3-3 et SP3-4. Le détail des fonctions de MuBu est décrit au § 2.2.5 p.98.

Dans la dernière période du projet, l'intégration Max/MSP de la bibliothèque a été finalisée. Les dépendances de la bibliothèque ont été réduites aux bibliothèques standard, sa stabilité dans des environnements multi-tâches a été assurée et la performance des accès aux données a été améliorée.

[Schnell09] Schnell N., Röbel A., Schwarz D., Peeters G., Borghesi R., « MuBu & Friends - Assembling Tools for Content-Based Real-Time Interactive Audio Processing in Max/MSP », *Proceedings of the International Computer Music Conference (ICMC)*, Montreal, CA, août 2009.

1.4.5.3 Morphing sonore et transformations temporelles interactives

Dans la première période du projet, le prototype d'une application de morphing sonore a été réalisé et présenté (voir D-SP3-5-1 en M10). Ce prototype intègre déjà la transformation temporelle avec l'échange et l'interpolation des propriétés sonores (« morphing ») entre deux sons. Pour la version finale d'une application prototype l'intégration de ces deux objectifs a été maintenue. Le détail des fonctions correspondantes est décrit au §2.2.5.2 p.101.

Pour le traitement des fichiers monophoniques, une deuxième version du module a été développée. Ce module permet aussi de transformer la hauteur des fichiers synchronisés pour adapter un fichier à la hauteur de l'autre ou pour créer un développement hybride de la hauteur à partir des deux fichiers. Dans ce processus, la hauteur moyenne d'un segment est distinguée de la modulation à court terme de la hauteur pour pouvoir traiter (et hybrider) les deux composants de manière indépendante.

1.5 Echantillonneur

Moteur Audio UVI (Universal Virtual Instrument)

Tout d'abord spécialisé dans la réalisation et l'édition de banques de sons pour échantillonneur matériel, la société Univers-Sons s'est lancée dans la création d'instruments virtuels depuis maintenant 8 ans. Alors pionnière dans le domaine, elle à décidé de créer un moteur audio permettant la relecture de banque de sons créés spécifiquement afin d'apporter une réelle plus value sur la simple vente de CD de samples. C'est ainsi que la première génération du moteur UVI est née.

Cette première mouture offrait déjà des fonctionnalités quasi comparables aux échantillonneurs matériels du moment avec la lecture d'échantillons sonores, application d'enveloppes, filtre et distorsion permettant de sculpter les sons selon ses désirs. Tout d'abord conçu comme un système monolithique en raison des limitations assez nettes des CPUs de l'époque, le moteur UVI à su évoluer afin d'enrichir les possibilités sonores de la solution. Cette évolution est tout d'abord passée par l'ajout d'effets sonores temps réel comme une réverbération puis de nombreux autres algorithmes de traitement ont suivi.

Néanmoins la source sonore est resté très longtemps limité à la lecture basique d'échantillon sonore jusqu'à la parution de MachFive 2 édité par la société américaine MOTU et de la 3ème génération du moteur UVI qui a enrichi sa palette sonore à l'aide d'un oscillateur type analogique basé sur des formes d'ondes simple. L'idée de Sample Orchestrator est donc venue de cette envie de pousser ce concept encore plus loin et d'étendre les possibilités sonores des instruments virtuels basés sur le moteur UVI.

1.5.1 Analyse des besoins et étude de marché

1.5.1.1 Analyse des besoins

Le projet Sample Orchestrator vise à créer un outil de sound-design d'un genre nouveau tout en permettant une utilisation standard de sampler ne le limitant ainsi pas à un usage exotique. Le but étant de toucher un maximum de clients potentiels, ne l'oublions pas.

Cet outil doit pouvoir être utilisé dans un cadre standard de production et post-production musicale avec des opérateurs allant du musicien au monteur son et se doit donc de remplir des critères d'interopérabilité avec le matériel couramment utilisé dans ces domaines.

Qu'il s'agisse de créer des textures sonores, de déclencher des bruitages ou tout simplement de composer une musique, SampleOrchestrator se doit de fournir un environnement de travail exemplaire et intuitif.

Tout d'abord, un état de l'art sur les logiciels et matériel utilisés dans ces milieux à été réalisé. Une liste de 14 produits ayant étant jugés représentatifs d'une utilisation typique a ainsi été sélectionnée. Pour chaque produit, une analyse des types d'utilisations ainsi que leurs points positifs et négatifs a ensuite été réalisée. Nous avons également testé en détail certaines de ces applications qui nous semblaient être les plus proches de Sample Orchestrator en terme de concept afin de noter les bonnes idées d'ergonomie. En général chaque produit est très orienté vers une palette sonore très spécifique ce qui peut laisser entendre qu'une place pour une application plus transversale comme Sample Orchestrator est disponible.

Une fois cette ligne directrice définie, une analyse précise des pré-requis nécessaires à la réalisation de ce projet à été mise en œuvre aidé en partie par l'état de l'art et notre propre expérience du domaine.

D'un point de vue technique, il est nécessaire que cette application puisse fonctionner sous forme de plug-in des principales stations de travail digitales du marché permettant une utilisation rapide et aisée dans un environnement où les acteurs ont souvent des emplois du temps très chargés. Le support multi-canal est devenu également indispensable dans le cadre de la création sonore au cinéma ou d'installation multimédia où les format 5.1 7.1 sont notamment très répandus.

Afin d'affirmer le caractère unique de cet outil, il se doit de fournir une large palette sonore couvrant la plupart des besoins de sound-design au sein d'une même interface de travail facilitant ainsi l'ergonomie et permettant d'explorer plus rapidement le champ sonore disponible.

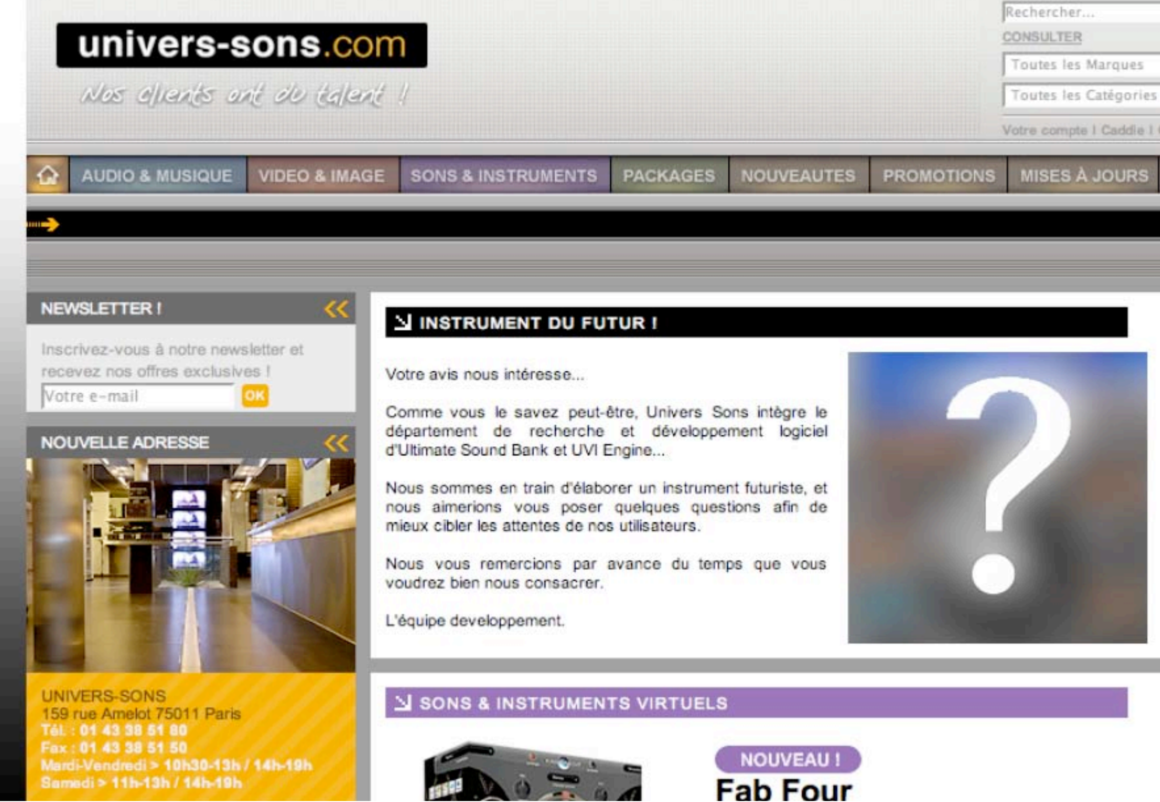
D'un point de vue ergonomique, Il s'agit de garder un logiciel unique avec un maximum d'information dans la page principale afin d'éviter les allers retours qui nuisent à la création.

L'esthétique du produit fait elle aussi partie du cahier des charges en raison de l'impact important que celle-ci à sur les clients potentiels en terme d'achat ou tout simplement sous forme de confort d'utilisation.

1.5.1.2 Etude de marché

Afin d'affiner les directions vers lesquels le projet Sample Orchestrator allait, il était nécessaire d'effectuer une étude de marché. Celle ci à été réalisé sous deux formes bien distinctes:

- Par le biais d'un sondage sur le site univers-sons.com, site du magasin de musique parisien, un des plus importants de France.



- Par le biais d'entretiens avec des acteurs « types » des différents marchés visés par l'application résultant du projet Sample Orchestrator .

Le sondage à porté principalement sur les attentes des utilisateurs en terme de design, d'usage et de fonctionnalités recherchées avec des questions d'ordre général. La disponibilité de question ouverte à permis d'obtenir également un avis plus personnel des personnes interrogées et des idées intéressantes et envisageables.

Les principales attentes des personnes ayant répondu aux sondages se sont portées sur la qualité sonore du produit ainsi que sur sa facilité d'utilisation.

Les entretiens ont porté sur des aspects beaucoup plus pragmatiques et spécifiques concernant la création sonore et ont permis une discussion sur les propositions que nous avions déjà envisagées pour résoudre certains problèmes que rencontrent fréquemment les personnes interviewées. Là aussi, la facilité d'utilisation à été notée comme étant un point important.

1.5.1.3 Conclusion

Ces différentes études ont permis de nous rassurer concernant les idées fondatrices du projet originel et nous a permis d'envisager des pistes nouvelles concernant l'ergonomie du produit.

L'état de l'art effectué lors de l'analyse des besoins, nous a également permis de noter que la majorité de ces logiciels mettent en valeur les possibilités de faire évoluer les sons au cours du temps, que ce soit de manière spatiale ou sur d’autres paramètres du son. Ce type de fonctionnalité est primordial si l’on veut rendre les sonorités vivantes, agréables voir plus « humaines ».

Le point de vue de l'ergonomie est également très important et même primordial. L'analyse et l'utilisation des différents outils du marché nous ont permis de nous faire une bonne idée de « ce qui marche et ce qui ne marche pas » et nous a permis d'effectuer une synthèse en sachant garder les points intéressants que nous avions rencontrés lors de nos tests.

1.5.2 Spécification fonctionnelle

Les étapes précédentes nous ont permis de définir un cahier des charges plus détaillé que le projet originel mettant en avant les points techniques définissant celui-ci. Nous allons ici en rappeler les principaux points:

- Application disponible sous forme de plugin et en version autonome pour Windows et OSX
- Support surround
- Possibilité d'organiser un arbre de synthèse afin de répartir sur un clavier différentes sources sonores générées par des « oscillateurs »
  - Définition de zone (Keygroups) et association d'un oscillateur à cette zone.

  - Mapping de ces zones en fonction de notes MIDI et de vélocités.
  - Nombreux type d'oscillateurs ayant dans certains cas une base d'échantillon sonore interchangeable
  - Association d'effets/filtres à ces zones.
  - Possibilité d'organiser ces zones en groupes (Layers).
  - Définition de règle d'activation de ces groupes en fonction de contrôle de jeux
  - Modulation des différents paramètres de synthèses par des sources de modulations.
  - Nombreux types de source de modulation ayant dans certains cas une base d'analyse de matériel sonore.
- Édition de forme d'onde et application de traitements
- Explorateur de banque de sons et recherche de matériel sonore
- Chargement et sauvegarde d'un grand nombre de format de sample (WAV, AIFF, SD2, SND,...)

On peut définir la partie échantillonneur du projet comme un méta-synthétiseur offrant les fonctionnalités d'un sampler évolué traditionnel tout en permettant l'utilisation de sources sonores plus souvent disponibles dans des logiciels expérimentaux alliées à un riche panel de transformations par le biais d'effets audio et de modulations de paramètres temporels.

Une synthèse de deux mondes qui sont habituellement assez séparés. Il s'agit ici d'offrir un outil poussé sans pour autant complexifier l'utilisation et à l'opposé sans brider le champ d'action.

1.5.3 Refonte du moteur UVI

Forts de notre expérience sur les versions précédentes du moteur UVI, nous avons décidé de pousser le concept plus loin et d'offrir une architecture plus ouverte et plus souple permettant une grande extension des fonctionnalités originelles.

La multitude de sources sonores, de modulations et de traitements à donc été au cœur de la conception de la nouvelle génération. Une attention toute particulière à été portée aux modulations temps réel des différents paramètres de contrôle en raison des informations amenés par les différentes études préalables. Il devait être possible de moduler les paramètres des différents modules par une ou plusieurs sources de contrôle interne ou externe à l'application.

Certains points techniques ont été également au centre des considérations, principalement liés à la gestion du coût de calcul entrainant la reconsidération de certains choix de conception. Un moteur par trop modulaire mène ainsi rapidement à un coût de calcul prohibitif pour notre cadre applicatif. Il s'agissait donc de trouver un compromis satisfaisant entre le coût de calcul et une modularité accrue, tout en conservant au minimum les performances originales.

Une compatibilité avec les banques de sons existantes était également nécessaire afin d'avoir une base de données conséquente lors de la sortie du produit et également d'offrir une transition simplifiée pour des utilisateurs de logiciels concurrents.

Un soin particulier à été apporté à la compatibilité avec les deux plateformes supportées (Mac, Windows) ainsi que les passerelle vers les différents formats de plugins (VST, AU,

MAS, RTAS). L'interaction entre l'interface graphique et le moteur audio à été renforcée afin d'offrir un système flexible de contrôle des paramètres du moteur audio et de faciliter le développement des interfaces utilisateurs relativement nombreuses.

1.5.4 Prototype d'interface de recherche

Une partie importante du projet Sample Orchestrator consiste à trouver rapidement et efficacement un matériel sonore pour servir de base à une création. Plusieurs modèles et stratégies de recherches ont ainsi été envisagés, du plus simple comme la recherche textuelle, au plus compliqué dans le cas de la recherche interactive par contenu. Dans le second cas, il s'agissait bien entendu de définir des critères pertinents quant à la perception et permettant de discriminer aisément les sons. La mise en forme de cette recherche et des résultats à été également au cœur des différents tests mis en œuvre afin d'offrir la meilleure expérience utilisateur.

Le premier prototype avait pour but de valider les choix techniques en terme de stockage de données et donc du type de la base de donnée choisie.

Le prototype suivant s'est orienté sur la partie graphique et la manière de représenter les résultats. Une fois les premières livraisons des algorithmes d'analyse de contenu fournis par l'IRCAM, les premiers tests de recherches par contenu ont pu être mis en place.

Ces 3 premiers prototypes ont permis de valider les concepts globaux de la recherche, et c'est donc sur l'ergonomie que se sont tourné les prototypes suivant afin de fournir à l'utilisateur des interface de recherches les plus adéquates et un affichage des résultats le plus pertinent.

1.5.5 Premier prototype Echantillonneur

Le but du premier prototype de la partie échantillonneur était de valider le bon fonctionnement du nouveau moteur de synthèse UVI; dans un premier en temps en validant des fonctionnalités comparable aux versions précédentes du moteur telles que: la lecture d'échantillons sonores, des modulations simples, un mapping de différentes sources sonores en fonction de paramètres de jeu (pitch, velocité) ainsi que la compatibilité et l'import des instruments existants.

Les principaux tests de validation se sont portés sur la fidélité du moteur audio d'un point de vue sonore et son exactitude concernant les temps d'enveloppes et des différentes sources de modulations. Afin de vérifier que le moteur était précis à l'échantillon près, des tests de rendu offline avec pour entrée des séquences préenregistré ont été effectués.

La stabilité et la robustesse de fonctionnement ont fait l'objet de nombreux tests étant données les conditions d'utilisation intensive des produits qui peuvent rester en fonctionnement pendant des mois.

Étant donnée la complexité des possibilités lors de l'utilisation temps réel du logiciel, des tests d'écoute A/B ont été réalisés avec des anciens produits afin de valider un rendu similaire aux versions précédentes du moteur.

L'interaction entre la partie graphique et le moteur audio était également un point important et de nombreux tests concernant la synchronisation des données entre les deux ont été nécessaires. Les paramètres pouvant être également édités à distance à l'aide d'un contrôleur MIDI, il fallait bien entendu que l'interface graphique reflète ces changements.

1.5.6 Prototypes Echantillonneur évolués

Les prototypes suivants ont eu pour objectif d'intégrer et de mettre en situation les différentes techniques de synthèse développées par l'IRCAM ainsi que les nouveaux modules développées au sein d'Univers Sons pendant toute la durée du projet.

Le moteur de TimeStrech/PitchShift SuperVP à été intégré sous forme d'un oscillateur permettant le chargement d'un échantillon sonore ainsi qu'un contrôle temps réel des paramètres de hauteur de son et de vitesse de lecture. Les préférences d'analyse sont également disponibles afin d'affiner les résultats de manière qualitative. La partie synthèse granulaire à travers l'API ZsaZsa à fait l'objet de deux oscillateurs dédiés pour chacune des applications potentiels.

Ces intégrations avaient pour but de détecter d'éventuels problèmes lors de la conception du nouveau moteur tout en permettant de vérifier l'adéquation des techniques mises en œuvre aux contraintes liés à l'application sampler. Notamment en ce qui concerne le coût moyen et maximum de la charge de calcul ainsi que l'adéquation des modules fournies. Leur utilisation au sein de l'application a également permis d'affiner la conception de l'interface utilisateur.

L'intégration des oscillateurs fournis par l'IRCAM avait également pour but d'affiner leur utilisation afin de les rendre plus « musicaux » dans le type d'utilisation fournis par Sample Orchestrator.

En supplément des modules fournis par l'IRCAM, de nouveaux composants rajoutés au moteur UVI ont put être testés et validés à l'aide des différents prototypes successifs. Les tests ont porté sur les qualités sonore des oscillateurs rajoutés ainsi que leur versatilité par rapport à des produits concurrents généralement plus dédié.

La lecture direct des échantillon sonore depuis le disque dur ainsi qu'un large panel d'effets temps réel ont fait partis des ajouts successif de fonctionnalité et donc de nouveaux modules à tester en détail.

D'un point de vue graphique, l'affichage graphique des modulations temps-réel à été réalisé et a permis également de valider visuellement le comportement de celle-ci: l'interaction entre modulations étant en général beaucoup plus facile a comprendre visuellement que théoriquement.

1.5.7 Interface utilisateur

L'étude préalable de différentes applications dans le domaine musical et dans des domaines connexes tels que la vidéo ou le graphisme, à permis d'inspirer et de dessiner les contours d'une interface graphique à la fois intuitive et professionnelle. Une prise en main rapide tout en permettant le travail quotidien des utilisateurs aguerris était au centre des discussions. L'effort c'est concentré sur la possibilité de maximiser l'espace graphique utile en présentant un grand nombre informations contextuelles tout en essayant de garder en permanence la vue d'ensemble au sein d'une seule et unique fenêtre consolidée.

Pour ce faire les points importants consistaient à pouvoir reconfigurer l'espace de travail en fonction de la tâche à accomplir, ainsi que de rendre l'interface entièrement redimensionnable afin de pouvoir bénéficier de toute la surface de l'écran disponible si nécessaire.

La partie purement visuelle, liée non pas au contenu mais à sa présentation formelle, est un point important et l'attrait graphique du produit est essentiel pour attirer l'œil des potentiels clients tout en évitant de fatiguer ou de déconcentrer les utilisateurs réguliers offrant aussi un guide de lecture de l'application. La conception graphique à donc été confié à un graphiste indépendant expérimenté dans le design de produits liés à la musique et notamment d'instruments virtuels et coutumiers de nos propres produits.

Les premiers prototypes se sont focalisés sur une vision très proche du fonctionnement interne du moteur en se rapprochant au fur et à mesure avec une vision pratique de l'utilisation

Plusieurs ébauches ont ainsi vu le jour et ce travail se peaufine encore également aujourd'hui.

1.6 Aide à l'orchestration

1.6.1 SP4.1 – Technologie de descripteurs audio pour l'orchestration

Objectifs : maîtriser les technologies relatives aux descripteurs audio utilisables par le moteur d'aide à l'orchestration

1.6.1.2 D-SP4-1-1 : Calcul de descripteurs

Responsable : Ircam-AnaSyn

Un ensemble de descripteurs des échantillons audio adaptés au problème de l'orchestration a été choisi. Certains de ces descripteurs sont dédiés aux sons monodiques pour décrire les sons de la base de d'échantillons, d'autre s'appliquent à des sons polyphoniques et sont utilisés pour décrire la cible. Ces descripteurs sont extraits par le programme ircamdescriptor écrit en Matlab, par les programmes C++ pm2 et supervp, ainsi que par des programmes dédiés, écrit en Matlab. Les fonctions permettant le calcul de ces descripteurs sur des grandes bases d'échantillons et leur organisation en une structure de données ont aussi été développées en Matlab. Les performances de ces descripteurs ont été testées dans des taches de classification des instruments et des modes de jeu.

Publications :

D. Tardieu, G. Peeters, X. Rodet. An Instrument Timbre Model For Computer Aided Orchestration. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA, New Paltz, New York, 2007.

D. Tardieu, G. Carpentier, X.Rodet. Computer-Aided Orchestration Based on Probabilistic Instruments Models And Genetic Exploration International Computer Music Conference, ICMC, Copenhagen, 2007

1.6.1.3 D-SP4-1-2 : Méthodes d'agrégation de descripteurs audio

Responsable : Ircam-AnaSyn

Les méthodes de calcul permettant, à partir des descripteurs de sons individuels, de prédire les valeurs de descripteurs correspondantes pour le son résultant de leur mixage dans des conditions d'orchestre ont été développées en Matlab. Elles sont basées sur des études de perception des sons, sur des principes d'acoustique ou, si nécessaire, sur des résultats empiriques Ces méthodes d'agrégation sont en partie utilisées dans le prototype 1 (SP4-2).

1.6.1.4 D-SP4-1-3 : Méthodes de distance à la cible

Responsable : Ircam-AnaSyn

Plusieurs méthodes de calcul de la distance entre une combinaison et la cible ont été proposées. Le but de ces méthodes est de donner une estimation de la similarité perçue entre la cible et la combinaison, elles sont donc basées sur des études de perception des sons et sur des principes d'acoustique. Ces méthodes, développées en Matlab, sont en partie utilisées dans le prototype 1 (SP4-2).

1.6.2 SP4-2 : Prototype expérimental d'orchestration 1

Objectifs : donner aux experts musicaux à un stade précoce du projet un prototype de moteur d'orchestration pour recueillir leur feed-back.

Partenaire/équipe responsable : analyse-synthèse / représentations musicales

1.6.2.2 D-SP4-2-1 : prototype 1 d'aide à l'orchestration

Responsable : Ircam-AnaSyn et Ircam-RepMus

Objet : donner aux experts musicaux à un stade précoce du projet un prototype de moteur d'orchestration pour recueillir leur feedback.

Description fonctionnelle :

- La connaissance instrumentale est réduite à une base de données instrumentales réduite, comprenant environ 5000 échantillons. La base se limite aux sons harmoniques, entretenus et sans variations temporelles.
- La description retenue se limite à deux descripteurs spectraux, les partiels les plus importants et le centroïde spectral.
- Une interface graphique permet de construire une cible à orchestrer à partir d'un son pré-enregistré, d'extraire et d'éditer ses descripteurs. Elle permet de procéder à une analyse en fondamentale multiple afin de filtrer la base pour hauteurs. Ce filtrage s'ajoute à un filtrage par instruments induit par la spécification d'un orchestre dans une interface ad-hoc. Une dernière interface permet d'explorer l'ensemble des solutions retournées par l'algorithme.
- Ce prototype utilise un algorithme d'exploration exhaustif uni-objectif. La distance à optimiser est calculée par agrégation hiérarchique des distances selon chaque descripteur.

Implémentation :

- Code Matlab
- Interfaces graphiques Matlab

Résultats / Performances :

- Le prototype permet de trouver des orchestrations pour de petits orchestres. Des expérimentations avec un compositeur ont permis d'imiter avec succès le son d'un instrument donné avec un petit groupe d'instruments.
- Le prototype permet également d'évaluer le type et le degré d'intéraction avec l'utilisateur.

1.6.3 SP4-3 : Modélisation de la recherche combinatoire

Objectifs : étude des méthodes de recherche et d'optimisation combinatoire ada ptées au problème de l'orchestration

Responsable : : Ircam-RepMus

1.6.3.2 D-SP4-3-1 : module sac-à-dos

Responsable : Ircam-RepMus

Objet : recherche de mixtures instrumentales solutions de l'orchestration par la technique de sac-à-dos multi-objectifs

Description fonctionnelle :

- La connaissance instrumentale est réduite à une base de données instrumentales réduite, comprenant environ 5000 échantillons. La base se limite aux sons harmoniques, entretenus et sans variations temporelles.
- La description retenue se limite à deux descripteurs spectraux, les partiels les plus importants et le centroïde spectral.
- Une modélisation du l'espace de recherche sous forme de chaînes binaires ainsi que la définition de fonctions de distances associées à chaque descripteur ont permis de poser le problème de l'orchestration comme une variante du sac à dos multidimensionnel multi-objectifs.
- L'approche retenue pour la résolution de ce problème en temps raisonnable utilise une population de solutions ainsi que des opérateurs génétiques de croisement et de mutation adaptés au caractère fortement contraint de l'espace de recherche. La gestion des contraintes

est « radicale », les solutions inconsistantes sont systématiquement éliminées de la population.

- Une recherche multi-objectifs permet d'adresser de façon pertinente, à l'aide d'un modèle à poids aléatoires, le caractère multidimensionnel de la perception du timbre.

Implémentation : Code matlab

Résultats / Performances :

- Pas de limitations dans la taille des orchestrations.
- Obtention d'un ensemble de solutions optimales, représentatives de possibilités sonores de l'orchestre.
- Possibilité, grâce à un modèle à poids, d'inférer les préférences de l'utilisateur en matière de timbre, et de concentrer la recherche dans une direction privilégiée de l'espace.

Publications

Carpentier Gregoire, Tardieu Damien, Assayag Gérard, Rodet Xavier, Saint-James Emmanuel*, An Evolutionary Approach to Computer-Aided Orchestration*. EvoMUSART. Valence, Espagne : Avril 2007, vol. LNCS 4448, p. 488-497

1.6.4 D-SP4-1-4 – Modèles instrumentaux

Responsable : Ircam-AnaSyn

Une méthode de description des instruments de musique et de différentes techniques de jeu à été proposée. Cette méthode est basée sur une modélisation probabiliste des descripteurs présentés dans la sous-tâche 4-1-1 et des dépendances entre descripteurs et variables symboliques (hauteur, dynamique, technique de jeu). Pour cela un réseau de dépendance entre les variables est proposé. Une structure d'organisation des modes de jeu sous forme de mixture d'expert est aussi définie. La distribution des descripteurs, pour chaque instrument et chaque technique de jeu, est apprise séparément sur un ensemble d'échantillons issus de plusieurs bases, commerciales ou non, souvent utilisées dans la recherche sur l'identification des instruments de musique. Ces distributions sont ensuite utilisées pour estimer la distribution des descripteurs pour toutes les hauteurs et nuances pour chaque instrument et chaque mode de jeu. Enfin, ces distributions élémentaires sont agrégées par des méthodes issues de la théorie de la décision collective pour estimer la distribution des descripteurs pour un instrument à un mode de jeu déterminé. Ainsi, les distributions des descripteurs du violon seront agrégées à celle du vibrato pour obtenir les distributions du violon joué vibrato. Toutes ces distributions constituent finalement des modèles du timbre des instruments pour différents modes de jeu à toutes les hauteurs et nuances possibles. Ces modèles constituent la connaissance du système d'orchestration concernant les instruments. Les algorithmes utilisant cette méthode ont été réalisés en matlab. Les tests effectués consistent en des tâches de reconnaissance automatique d'instruments et de techniques de jeu. Les résultats obtenus montrent l'efficacité de la méthode. Au niveau des instruments, les taux de reconnaissance sont comparables à l'état de l'art. Au niveau des modes de jeu, certain comme le vibrato, le trémolo ou l'éolien sont très bien reconnus (90%), d'autre comme le sul-tasto sont moins bien reconnus car ils ont un timbre très similaire à d'autres modes de jeu.

Des méthodes d'addition de modèles ont ensuite été proposées. Elles permettent de trouver la distribution des descripteurs d'une mixture de son (une orchestration) à partir des distributions des sons composant la mixture. Pour chaque descripteur, la formule d'addition proposée dans la sous-tâche 4-1-2 est adaptée à des distributions gaussiennes en ayant parfois recours à des approximations (comme la méthode de Fenton-Wilkinson pour l'addition de loi log normale). Ces méthodes sont utilisées dans un programme matlab qui permet d'évaluer la pertinence d'une orchestration sachant la cible. Ce programme est testé dans des tâches de reconnaissance de mixture de sons d'instruments d'orchestre.

Ce programme sera intégré dans le système final et permettra d'obtenir un choix plus vaste de solutions. Les interfaces de communication entre les programmes de cette sous-tâche et

le programme d'orchestration ayant été prédéfinies, l'intégration des deux modules ne pose pas de problème technique particulier.

1.6.5 SP4-3 – Modélisation de la recherche combinatoire

1.6.5.1 D-SP4-3-2 – Module génétique

Responsable : Ircam-RepMus

Fonctionnalités réalisées :

L'approche retenue pour la recherche de mixtures instrumentales s'approchant de la cible sonore consiste en la génération et la modification d'une population de solutions selon des mécanismes inspirés de la biologie et de l'évolution des espèces. Dans ce type d'algorithmes, dits évolutionnaires, les meilleures solutions survivent et transmettent leurs caractéristiques aux générations suivantes, l'ensemble de la population évoluant ainsi vers des solutions efficientes. L'algorithme est dit génétique lorsque les solutions sont représentées par une chaîne de caractères ou d'entiers dont chaque élément décrit de façon unique et indépendante une caractéristique particulière de la solution. On parle alors de « gènes » s'exprimant dans l'espace des « décisions » (pour le compositeur, c'est la partition), et l'évolution de la population se fait au moyen d' « opérateurs génétiques » qui agissent sur un ou plusieurs gènes d'une configuration existante pour en engendrer une nouvelle dont certaines caractéristiques seront communes avec la ou les configuration(s) parente(s).

Les configurations sont représentées par un vecteur de couples variable/valeur, où chaque variable est associée à un instrument de l'orchestre, et dont les valeurs sont l'ensemble des sons possibles pour cet instrument. Cet ensemble dépend à la fois du problème courant et de la connaissance instrumentale globale du système. Cette représentation assure qu'en l'absence de contraintes autres que celles liées à l'effectif orchestral, toute solution instanciée aléatoirement est jouable par l'orchestre. Les opérateurs génétiques employés sont le croisement uniforme (uniform crossover) et la mutation par modification aléatoire d'un seul gène. En présence de contraintes, la probabilité de mutation est inversement proportionnelle à la difficulté du problème.

Implémentation : code Matlab.

Date de réalisation : mars 2008.

1.6.5.2 D-SP4-3-3 – Module recherche locale

Responsable : Ircam-RepMus

Fonctionnalités réalisées :

En plus des contraintes liées à l'effectif orchestral, un ensemble d'opérateurs permet d'exprimer un réseau de contraintes complexe que doivent vérifier les solutions d'un problème d'orchestration. Ces contraintes s'appliquent sur des descripteurs symboliques uniquement, qui sont les paramètres d'écriture pour le compositeur. L'utilisateur peut donc facilement imposer un certain nombre de restrictions sur les instruments utilisés, les familles, les notes, les sourdines, les dynamiques et les modes de jeu.

Le module de recherche locale permet, dans le cas d'un problème contraint, de « réparer » les solutions éventuellement inconsistantes. L'algorithme CDCSolver tire parti de la distinction entre « contraintes de design » et « contraintes de conflit » pour trouver une solution consistante en un nombre limité d'itérations. Cette dichotomie commande l'instanciation ou la désinstanciation d'une variable. Inspiré à la fois de l'algorithme de recherche adaptative (Truchet, Codognet & Diaz) et de l'heuristique CN-Tabou (Vasquez, Habet et Dupont), il s'articule autour d'une recherche tabou dynamique au cours de laquelle la plus mauvaise variable est modifiée en priorité. En outre, une gestion adéquate des instanciations partielles permet une propagation implicite des affectations.

Implémentation : code Matlab.

Date de réalisation : mars 2008.

Performances :

L'algorithme CDCSolver a été évalué sur 500 problèmes de petite taille pour lesquels les espaces de recherche et les ensembles de solutions sont connus. Cinq niveaux de contraintes ont été utilisés, réduisant progressivement l'ensemble des solutions consistantes de 8.78% à 0.01% de l'espace de recherche. Pour chaque niveau de contrainte le nombre d'itérations de CDCSolver est comparé est nombre moyen d'itérations au bout duquel on peut espérer générer aléatoirement une solution consistante (random). Les performances de CDCsolver par rapport au Random sont reportées dans le tableau suivant :

Niveau de contrainte	Nombre d'itérations	Random	Rapport au random
1	5.0	12.2	2.6
2	19.4	403	26.0
3	41.1	1115	32.1
4	38.3	2104	67.0
5	102.8	19469	264.1

1.6.6 SP4-4 – Prototype expérimental d'orchestration 2

1.6.6.1 D-SP4-4-1 – Prototype 2 d'aide à l'orchestration

Responsable : Ircam-RepMus

Fonctionnalités réalisées :

Un second prototype expérimental d'aide à l'orchestration a été réalisé, incorporant les méthodes d'optimisation combinatoire sous contraintes issues de SP-4-3. Les modèles d'instruments issus de SP-4-1, en cours de tests de validation, ne sont pas encore intégrés à ce prototype.

Développé en Matlab, cet outil offre une interface utilisateur spécifique à chaque étape du processus d'orchestration, de la construction d'un orchestre, la spécification de contrainte et l'analyse d'un son jusqu'à une navigation avancée dans l'espace des solutions efficientes, selon plusieurs points de vue (symbolique, espace de timbre, optimalité). La simulation des solutions est prise en charge par un sampler développé en Max/MSP et dialoguant via OSC avec le prototype. Une interface via OSC avec OpenMusic permet également la création de cibles abstraites (voir D-SP-4-5). Le prototype permet en outre de relancer la recherche à partir d'une solution intermédiaire tout en inférant les préférences implicites de l'utilisateur dans le choix de celle-ci, ainsi que de transformer progressivement une solution courante à l'aide de contraintes supplémentaires, offrant ainsi un premier contrôle sur l'évolution temporelle du timbre.

Ce prototype a été utilisé par le compositeur Jonathan Harvey pour l'écriture du 3ème mouvement de sa pièce « Speakings » créée à Londres en août 2008.

Implémentation : code Matlab.

Date de réalisation : juin 2008.

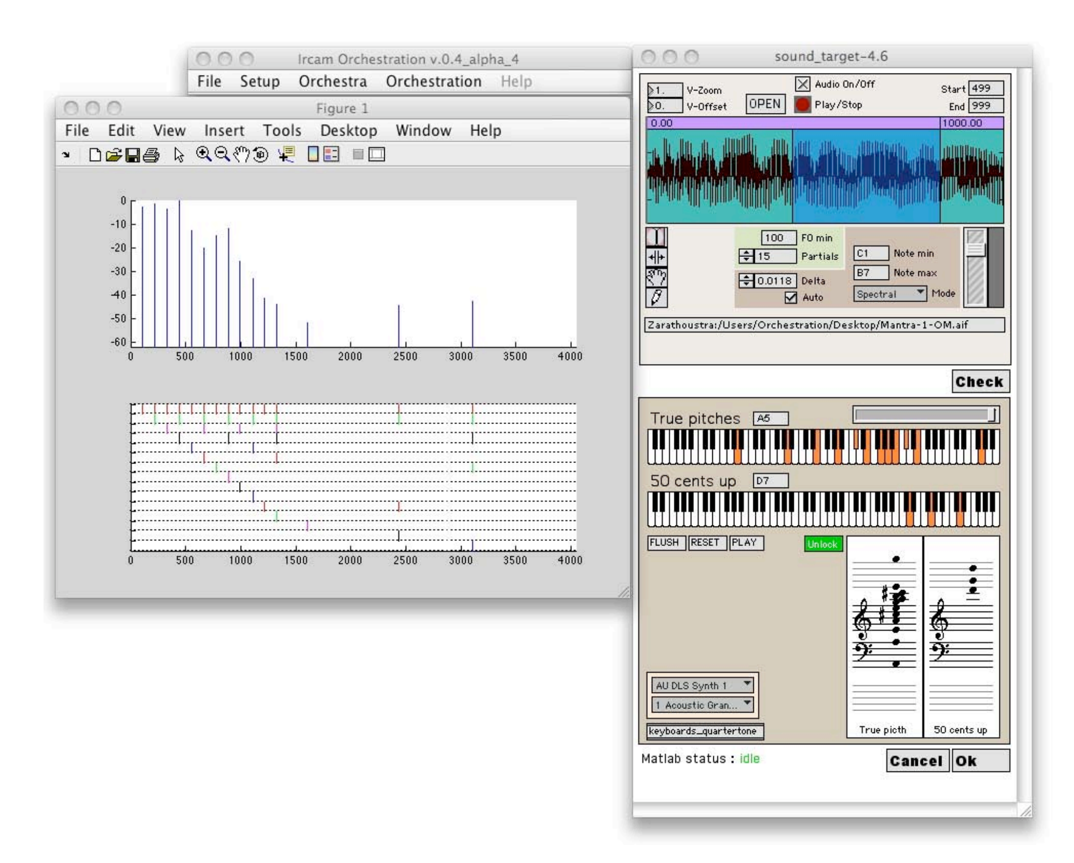


Figure 35 - Prototype 2 : Editeur de cible

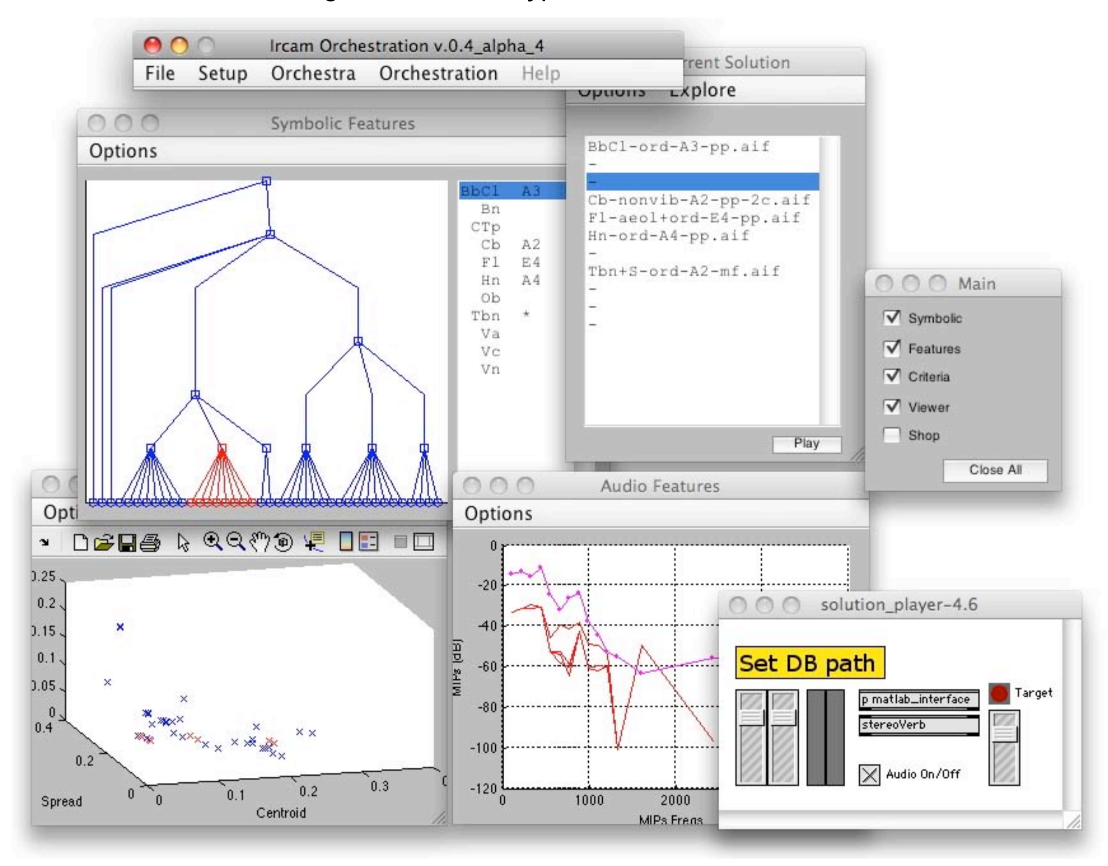


Figure 36 - Prototype 2 : Navigation dans les solutions

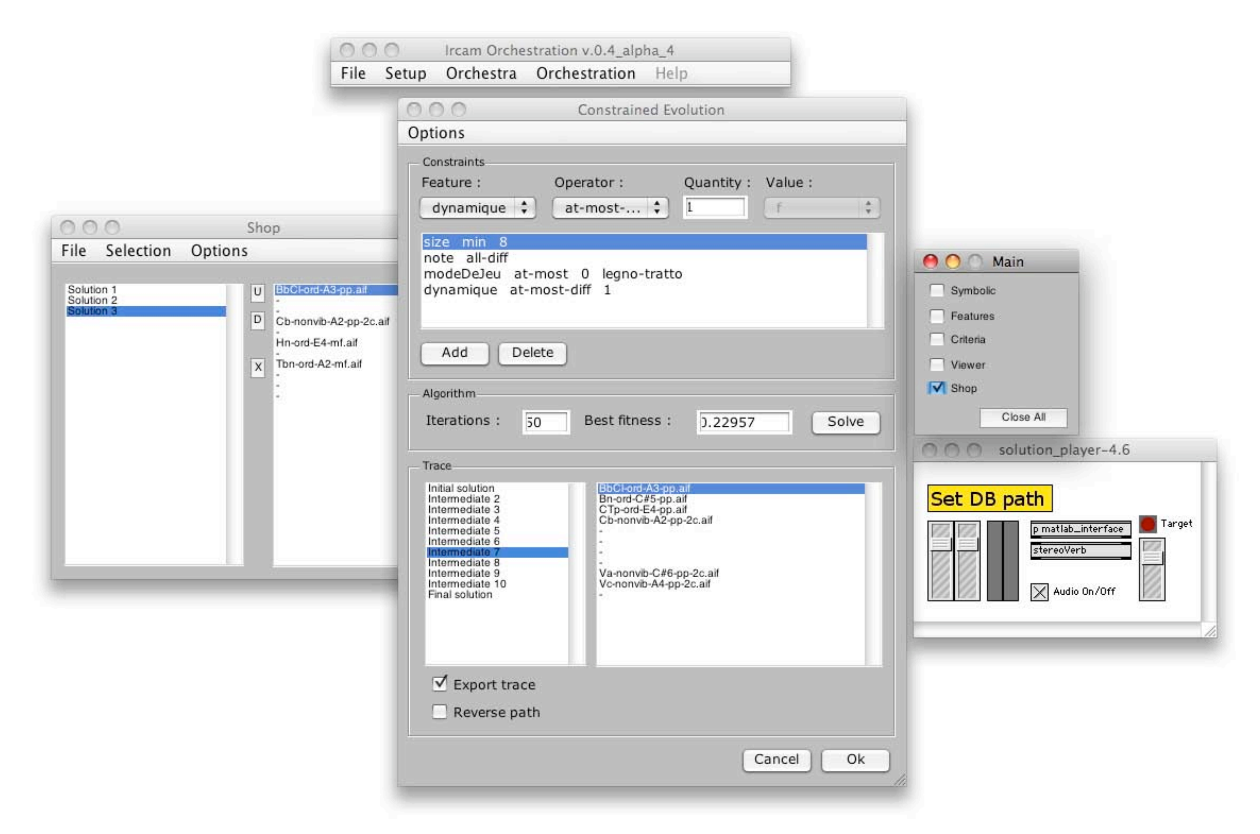


Figure 37 - Prototype 2 : Transformation temporelle par les contraintes

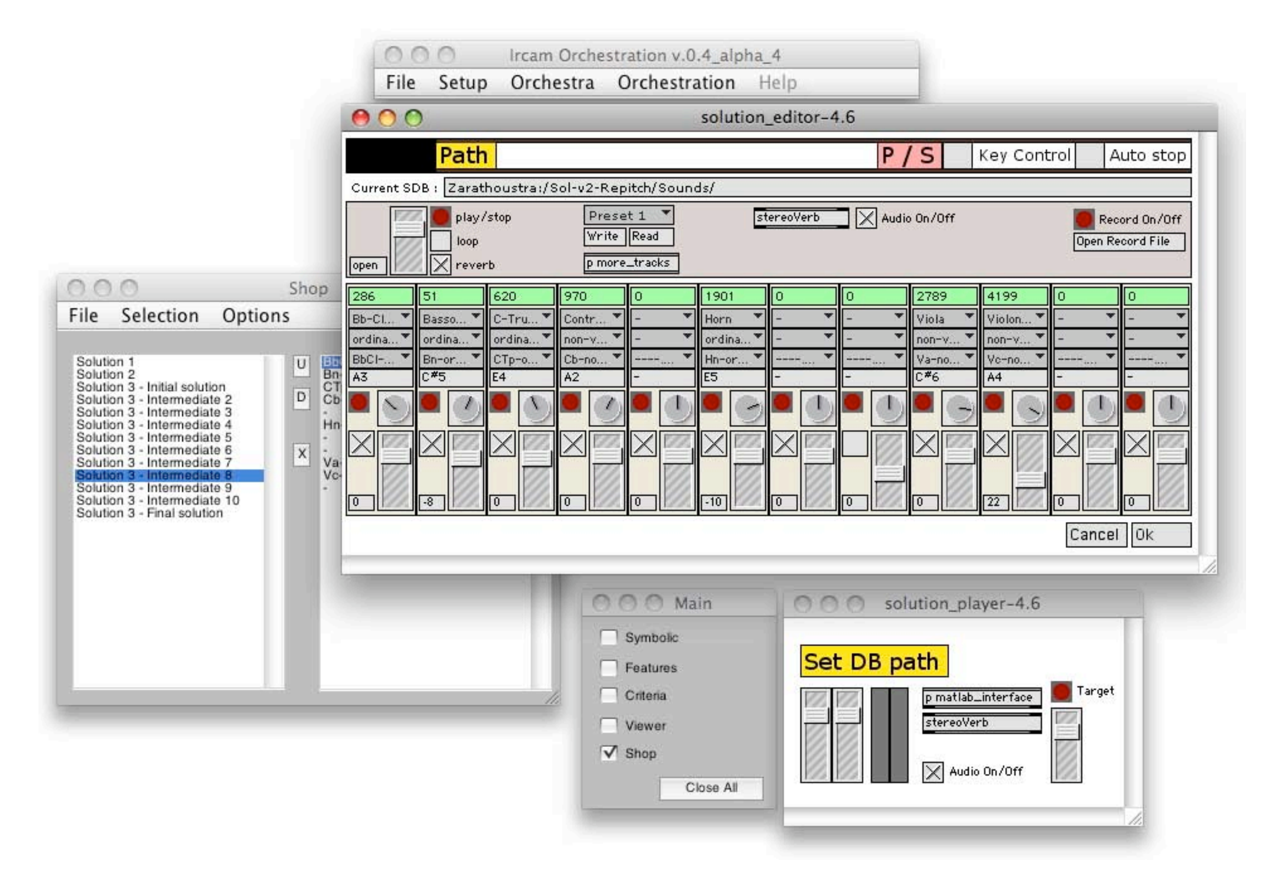


Figure 38 - Prototype 2 : Editeur de solutions

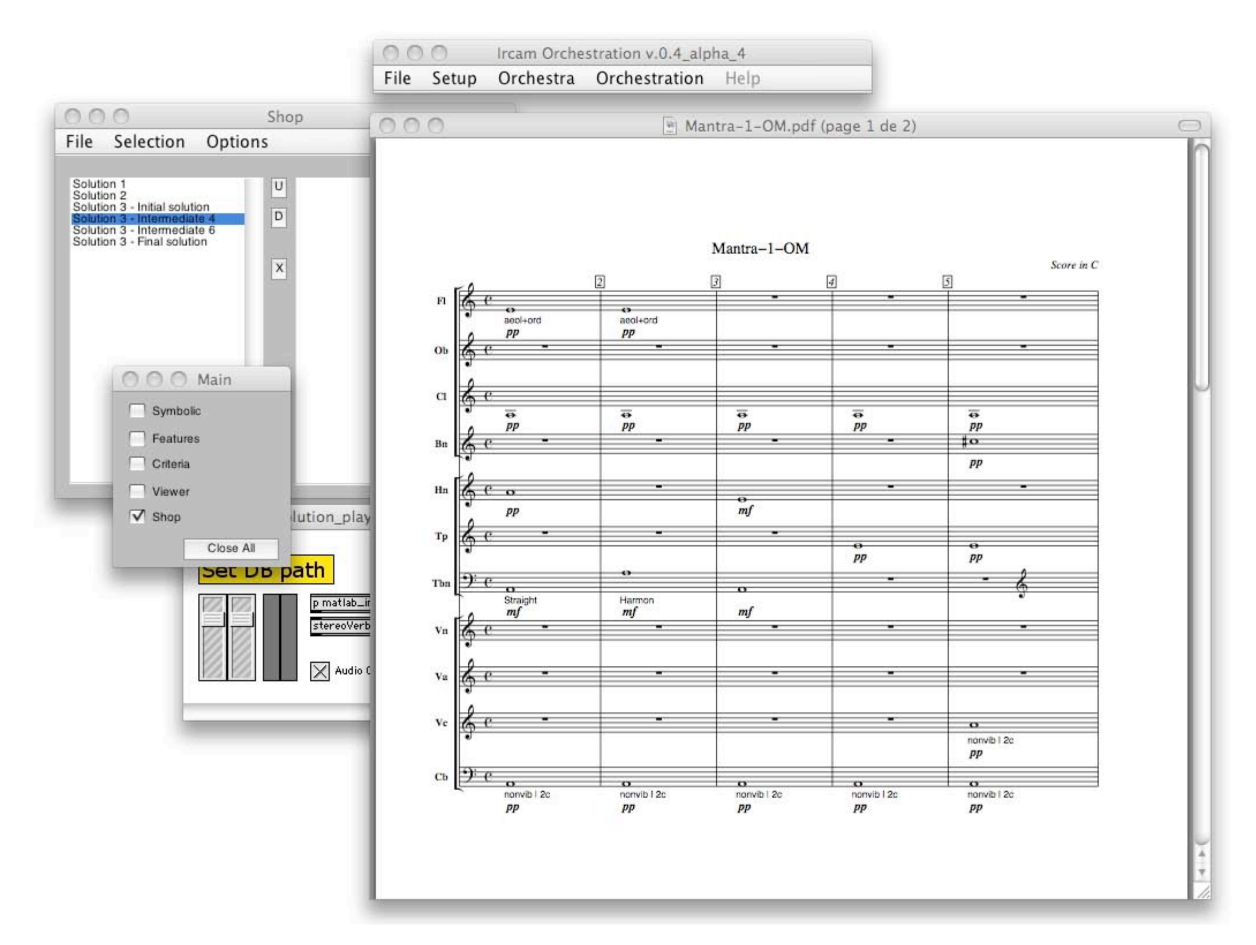


Figure 39 - Prototype 2 : Partition résultante

1.6.7 SP4-5 : création de cibles abstraites pour l'orchestration

Objectifs : interface de synthèse de cible abstraite
Responsable : Ircam-RepMus

1.6.7.2 D-SP4-5-1 : synthèse de cibles abstraites

Objet : environnement OpenMusic de synthèse élémentaire de cibles abstraites à partir de données symboliques (accords, séquences, patches de CAO), utilisant les technologies de synthèses disponibles dans l'équipe Analyse-synthèse.

Fonctionnalités réalisées :

La librairie Chroma de OpenMusic a été mise à profit pour réaliser les cibles de synthèse. Tout l'environnement de composition assistée dans OpenMusic peut ainsi être mis à profit pour composer des séquences musicales (principalement séquences harmoniques) qui servent de support à la synthèse de timbres complexes. Les données issues de la CAO sont converties en paramètres généraux de synthèse qui alimentent un objet Chroma, lequel peut ensuite alimenter plusieurs moteurs de synthèse ou de traitement disponibles (additive, svp, modalys). Les cibles ainsi constituées peuvent ensuite être soumises au solveur d'orchestration.

Implémentation : code Common-Lisp

Date de réalisation : Décembre 2007.

1.6.7.3 D-SP4-5-2 : éditeur de cibles abstraites

Objet : environnement OpenMusic de création, d'édition, de transformation, de cibles abstraites pour l'orchestration.

Fonctionnalités réalisées :

Cet éditeur se place entre les outils de CAO qui sont au niveau des paramètres classiques de hauteur, durée ou intensité, et le moteur de synthèse chroma, qui manipule des paramètres numériques de synthèse. Il permet à partir d'un objet musical tel que l'accord ou la séquence d'accord (et en théorie toute polyphonie) de proposer des enrichissements du niveau spectral, par l'affichage du spectre résultant, et par un ensemble d'opérateurs d'édition :ajout,suppression, enveloppe, etc.

Il permet de lancer la synthèse à travers le moteur Chroma et ensuite d'échanger des informations avec le prototype d'orchestration du SP-4-4-1 en mettant à profit le protocole OSC. Trois types d'informations peuvent être échangés : le fichier son résultant de la synthèse, les paramètres de la synthèse, et l'objet musical symbolique original. Les deux derniers peuvent être optionnellement mis à profit par le moteur d'orchestration pour contraindre la recherche, ou pour varier le type de synthèse lui-même de manière à contrôler certains des descripteurs utilisés pour la recherche (notamment les partiels significatifs).

Implémentation : code Common-Lisp

Date de réalisation : Mai 2008.

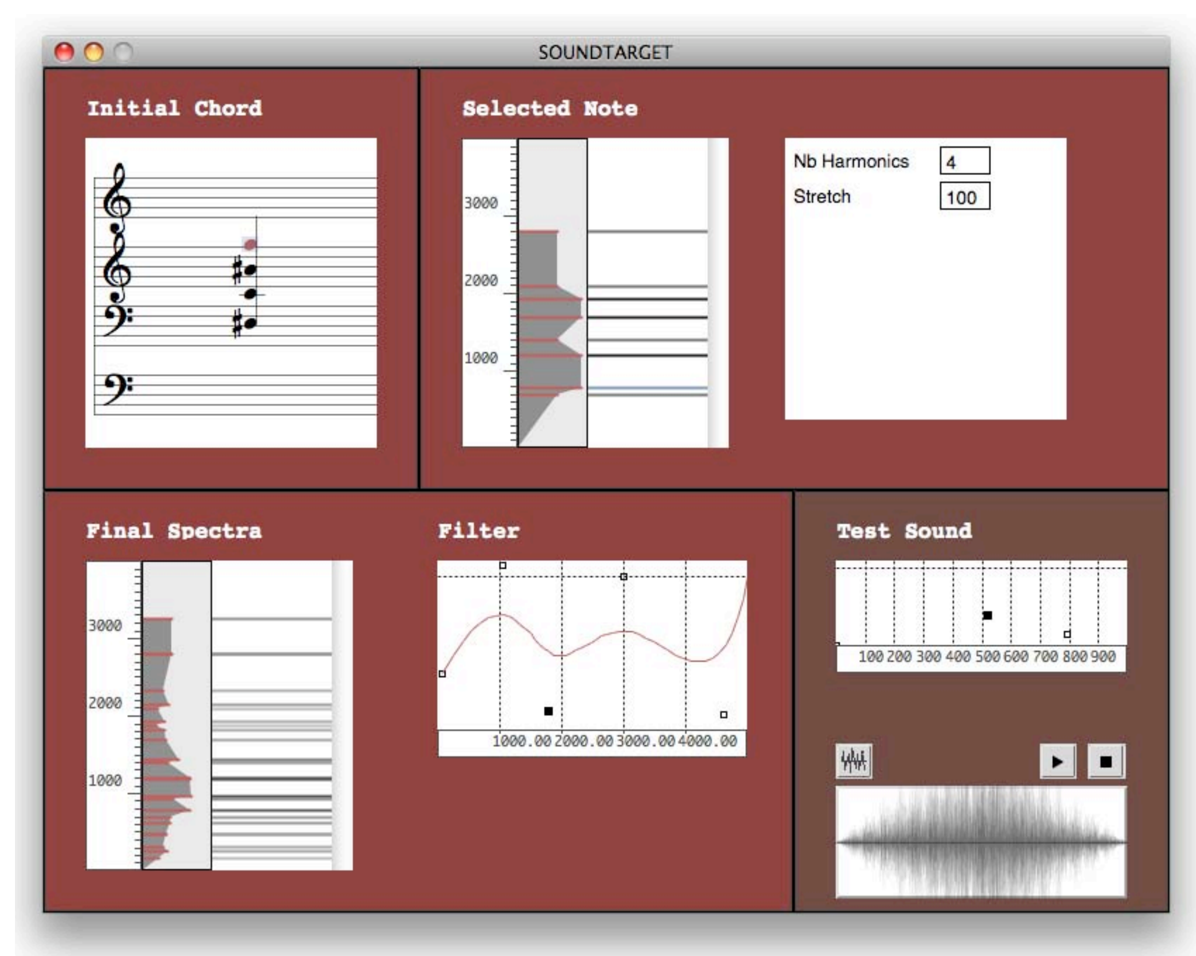


Figure 40 - Editeur de cibles dans OpenMusic

1.6.8 SP4-6 : environnement OM de contrôle de l'orchestration

Note : à partir de SP4-6, une prolongation a été accordée au projet. Nous avons profité de cette prolongation pour décider d'une architecture optimale en vue de l'utilisation concrète en production musicale du système d'orchestration. Dans cette perspective, le moteur d'orchestration (*Orchidée*) devient un serveur communicant avec ses clients grâce au protocole OSC (Open Sound Control). L'application OpenMusic devient ainsi un client d'orchestration (*OM-Orchidée*). D'autres clients sont prototypés à partir d'autres plates-formes logicielles (e.g. Max/MSP)

Objectifs : interface générale OpenMusic de contrôle de l'orchestration

1.6.8.1 D-SP4-6-1 : interface de contrôle de l'orchestration 10/2008

Objet : prototype I environnement OpenMusic de contrôle de l'orchestration articulant : la spécification de cibles concrètes, la création de cibles abstraites, le paramétrage et le lancement du moteur de résolution d'orchestrations, le choix des solutions.

<u>Interaction musicale : architecture client/serveur</u>

L'intégration du moteur *Orchidée* en relation avec un environnement compositionnel est conçue sur le modèle d'une architecture client / serveur. Le serveur, ou moteur de résolution, intègre les aspects calculatoires du système. Il communique avec une ou plusieurs interfaces clientes en relation à un environnement ou un processus compositionnel donné. Cette communication se décline selon deux perspectives complémentaires: spécification du problème et navigation dans l'espace de solutions.

La spécification du problème d'orchestration comprend elle-même trois aspects principaux : 1) la composition orchestrale, c'est-à-dire l'ensemble des "variables" du problème (instruments disponibles et configuration de l'orchestre) et des "valeurs" pouvant être prises par ces variables ; 2) une cible, décrivant de façon plus ou moins abstraite le résultat que l'on souhaite obtenir ; 3) des filtres et contraintes symboliques qui expriment des caractéristiques souhaitées ou imposées dans la résolution du problème et les solutions proposées (contraintes harmoniques, sur les modes de jeux, etc.) Un client Orchidée communicant avec le moteur de résolution a donc pour fonction initiale de communiquer ces différents types de données au serveur.

La question de la spécification de la cible mérite un intérêt particulier. Le serveur *Orchidée* reçoit celle-ci en tant que son "concret" (un fichier audio), associé le cas échéant à des information complémentaires, qu'il va analyser de sorte à constituer une "abstraction" sonore, ou une classe de son dont la résolution aura pour objectif de découvrir de nouvelles instances. On parlera de cible concrète ou abstraite, selon que cette cible ait été spécifiée de manière concrète à partir d'un son enregistré, ou de manière plus abstraite. Cet aspect sera repris dans la conception des interfaces client, notamment avec la librairie OM-Orchidée.

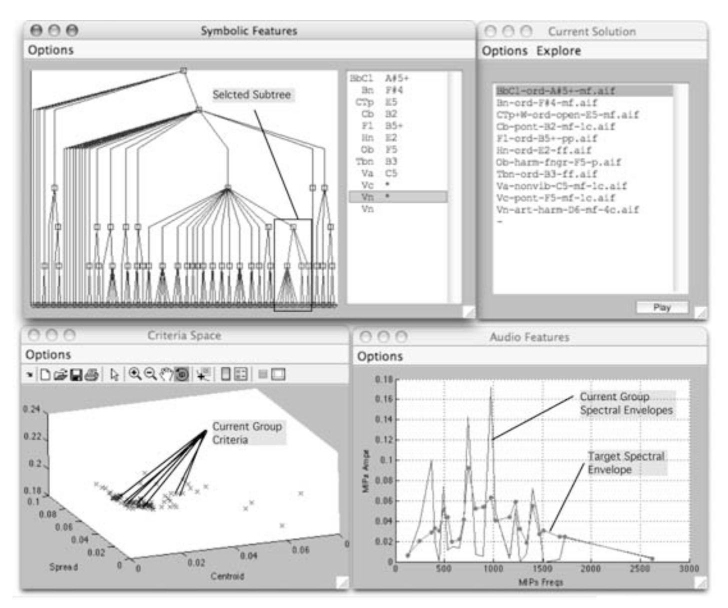
Suite à la spécification du problème, et étant donné un ensemble de solution proposées par le moteur de résolution, la navigation dans l'espace de ces solutions permettra alors de sélectionner, écouter, modifier certaines de ces solutions afin d'affiner la recherche, de progresser dans l'exploration de l'espace sonore et finalement de les intégrer dans un processus d'écriture. Les données et processus de résolution sont donc gérés et actualisés régulièrement par Orchidée en fonction des nouvelles données et requêtes communiquées par les clients: la prise en compte des préférences de l'utilisateur lors de la navigation intègre ces données pour les recherches ultérieures, de sorte que le compositeur participe implicitement ou explicitement aux différentes étapes de recherche et de calcul.

Concernant l'exploration des solutions, Orchidée met en avant une représentation simultanée des espaces de représentations symboliques, des descriptions sonores, et des critères de recherche. Chaque solution est caractérisée par un ensemble de variables

musicales dans l'espace symbolique (hauteur, dynamique, mode de jeu, etc.), mais aussi par un ensemble de descripteurs perceptifs, et par un ensemble de critères de recherche, ce qui permet d'envisager une navigation parallèle dans ces différents espaces. Les transformations et parcours de recherche et de navigation dans ces espaces constituent alors autant de possibilités d'exploration des solutions et permettent de structurer la recherche dans une boucle d'interaction avec le moteur de résolution.

Enfin un dernier aspect de l'interaction utilisateur concerne la gestion de la base de données et de connaissance instrumentale, avec un ensemble de requêtes et d'instructions permettant d'analyser une base d'échantillons sonores pour en extraire les descriptions et enrichir cette base, ou encore s'enquérir de ses différentes caractéristiques pour la spécifications ultérieure de filtres et contraintes de recherche.

Différentes interfaces on été réalisés à l'heure actuelle, s'inscrivant dans cette interaction avec le moteur d'orchestration. La première a été intégrée au prototype initial d'orchestration en MatLab, et concerne principalement l'exploration et la navigation dans l'espace des solutions. Les suivantes ont été réalisés dans Max/MSP et dans OpenMusic comme clients de l'application finale.



Navigation dans l'espace des solutions: prototype MatLab

<u>OM-Orchidée: une bibliothèque dans OpenMusic</u>

Le description de la version finale de l'interface cliente OM-Orchidée constitue l'un des résultats finaux du projet et est donnée au §2.4.2.

1.6.9 SP4-7 : tests/validation

Objectifs : tests de validation avec les compositeurs à diverses phases du projet

Durée: répartie sur la durée du projet.

Tout au long du projet nous avons bénéficié de l'expertise d'un groupe de compositeurs qui ont régulièrement donné du feed-back sur les prototypes présentés. Toutes les décisions stratégiques en termes d'architecture et d'ergonomie ont tenu compte de ce retour d'usage. Yan Maresz, Philippe Schoeller, Georges Bloch, Tristan Murail, Joshua Fineberg, Geoffroy Drouin, Michael Levinas entre autres ont interagi à un moment ou à un autre avec les chercheurs.

De manière à passer « l'épreuve du feu », le Prototype 2 a été utilisé en production musicale grandeur réelle par les compositeurs du cursus d'informatique musicale de l'Ircam et leurs pièces ont été jouées en concert public. A l'occasion de cette collaboration certaines demandes (orchestrations en micro-intervalles) ont été formulées et satisfaites.

Une grande production internationale (Speakings pour Orchestre de Jonathan Harvey, première aout 2008 à Londres au Royal Albert Hall, BBC Orchestra) a permis de vérifier que le travail de l'équipe arrivait à maturité. L apartition utilise uen grande quantité d'orchestrations directement issues de notre système.

1.6.10 SP4-8 : Application finale

1.6.10.1 D-SP4-8-1 : application finale jusqu'en juin 2009.

Objet : intégration des interfaces de contrôle, d'édition, de synthèse, de recherche dans l'application finale. Réécriture complète sous forme de serveur *Orchidée*.

Le description de la version finale du serveur Orchidée constitue l'un des résultats finaux du projet et est donnée au §2.4.1.

1.6.11 Orchestration musicale dynamique

Stage Atiam (Master Université Paris 6) de Philippe Esling de mars à juin 2009.

Ce stage, effectué au cours des derniers 4 mois du projet, avait pour but d'aborder une nouvelle dimension non encore abordée de l'outil d'orchestration : la v

En suivant la direction des travaux réalisés à l'Ircam concernant l'orchestration, pendant ce stage nous avons étendu ces recherches en nous concentrant sur la composante temporelle suivant les facettes micro et macro temporelles :

Grâce à la proposition et implémentation d'un modèle instrumental, nous pouvons prendre en compte aujourd'hui le timbre comme une structure spectro-temporelle. Celui-ci permet à la fois de modéliser de manière précise les propriétés spectrales des instruments mais également d'inférer la structure d'un son de durée modifiée. Nous avons proposé ainsi dans le cadre de ce stage, un modèle d'évolution temporelle des propriétés spectrales basé sur les mixtures de gaussiennes que nous avons comparé et validé par rapport à ceux existants dans la littérature. Ce modèle permet en outre de capturer les propriétés pertinentes des timbres instrumentaux avec une grande réduction dimensionnelle tout en permettant une resynthèse efficace de ceux-ci. De plus, le point fort de ce modèle est de permettre d'inférer de manière quasi automatique la structure temporelle d'un son identique mais de durée modifiée tout en conservant les segments d'attaque et de relâchement. Nos travaux s'inscrivant dans la continuité des développements entrepris à l'Ircam, nous présentons les résultats d'expériences issues de l'intégration de ce modèle dans le système Orchidée en notant la nette extension apportée au modèle statique.

Nous avons tenté dans un second temps d'utiliser une première approche pour aborder le problème du macro temps. De par la complexité relative d'un tel problème nous avons temporairement laissé le problème combinatoire en retrait. Nous avons donc choisi une première approche pour tenter d'étendre l'orchestration par un son cible en permettant à celui-ci d'être un morceau structuré au niveau de la forme musicale ayant différents éléments à orchestrer. En inversant le paradigme, l'orchestration musicale est considérée comme un problème de séparation de sources où le but est de trouver la combinaison d'instruments qui explique au mieux la « macro cible ». Compte tenu de cette approche, nous utilisons la

factorisation en matrices non-négatives (NMF) pour permettre de décomposer le spectre de la cible en une somme pondérée des instruments choisis. L'objectif est d'abord de découper le spectre en un ensemble d'unités temporelles qui ont une certaine cohérence spectrale. Cette technique permet à la fois de trouver le mélange instrumental 'optimal' pour un élément mais également d'envisager l'utilisation du système dans un cadre de réorchestration d'une musique entière. Ainsi grâce à la technique de factorisation de matrices non-négatives nous pouvons calculer rapidement la décomposition sur une base instrumentale. D'une part, l'ajout des contraintes de parcimonie nous permet de limiter l'effectif instrumental, d'autre part les contraintes de continuité locale nous permettent également d'ajouter une logique temporelle en pénalisant les propositions ayant de trop grandes variations instrumentales (à la fois de notes et de classes d'instruments).

La formulation d'une orchestration par l'utilisation d'un son cible peut être considérée comme un point d'entrée dans un domaine fermé à la formalisation. Le but de l'orchestration est généralement de produire et non reproduire des mélanges instrumentaux. Ainsi, une direction intéressante pour les travaux futurs serait de contourner cette approche “bas niveau” par la définition d'une cible abstraite pour le système. Le problème principal dans nos recherches sera de trouver un formalisme adéquat pour la réification de ce qui pourrait être une intention orchestrale. Après plusieurs entretiens avec des compositeurs, une approche intéressante est apparue dans la notion d'esquisse. Les compositeurs comptent généralement sur un premier croquis d'une pièce musicale. Ce schéma définit l'intention principale du compositeur et devrait offrir un cadre large d'investigation pour l'orchestration. Cependant il faut d'abord analyser des esquisses pour savoir à la fois si celles-ci ne relèvent pas d'un processus trop subjectif et en même temps comprendre à quel point le compositeur “pense l'orchestre”. La continuation logique de l'approche cible sonore serait donc la spécification d'une cible abstraite, utilisant tous les outils de la CAO pour essayer de converger vers une idée de timbre mais qui ne se réduit pas à une réalité sonore.

1.6.12 Bibliographie

[Carpentier08a] Carpentier, G., « Approche computationnelle de l'orchestration musicale : optimisation multicritère sous contraintes de combinaisons instrumentales dans de grandes banques de sons », Thése de doctorat Atiam, UPMC, Ircam 2008

[Carpentier07a] Carpentier, G., Tardieu, D., Assayag, G., Rodet, X., Saint-James, E., « An Evolutionary Approach to Computer-Aided Orchestration », EvoMUSART, vol. LNCS 4448, Valence, 2007, pp. 488-497

[Carpentier06a] Carpentier, G., Tardieu, D., Assayag, G., Rodet, X., Saint-James, E., « IMITATIVE AND GENERATIVE ORCHESTRATIONS USING PRE-ANALYSED SOUNDS DATABASES », SMC'06, Marseille, 2006, pp. 115-122

[Carpentier05a] Carpentier, G., « Information technology — Multimedia content description interface — Part 4: Audio, AMENDMENT 2: High-level descriptors », Motion Picture Expert Group (ISO/IEC JTC 1 SC29, Poznan/ Pologne, 2005

[Carpentier09a] Carpentier, G., Bresson, J., « Interacting with Symbolic, Sound and Feature Spaces in Orchidée, a Computer-Aided Orchestration Environment », Computer Music Journal, to appear 2009

[Carpentier09b] Carpentier, G., Tardieu, D., Harvey, J., Assayag, G., Saint James, E., « Constrained Multiobjective Optimization for Computer-Aided Orchestration: Theory and Applications », Journal of New Music Research, to appear 2009

[Carpentier09c] Carpentier, G., Assayag, G., Saint James, E., « Solving the Musical Orchestration Problem using Multiobjective Constrained Optimization with a Genetic Local Search Approach », Journal of Heuristics, to appear 2009

[Tardieu08b] Tardieu, D., Carpentier, G., Peeters, G., « Instrument sound description and modelisation in the context of computer aided orchestration », Acoustics 08, 2008

[Tardieu08c] Tardieu, D., « Modèles d'instrumentspour l'aide à l'orchestration », Thèse de doctorat Atiam / Paris 6, IRCAM-Paris 6, 2008

[Tardieu07a] Tardieu, D., Carpentier, G., Rodet, X., « Computer-Aided Orchestration Based on Probabilistic Instruments Models And Genetic Exploration », ICMC, Copenhagen/Denmark, 2007

[Tardieu07b] Tardieu, D., Rodet, X., « An Instrument Timbre Model For Computer Aided Orchestration », WASPAA, New Paltz, NY, 2007

G. Carpentier, *CDCSolver: Un algorithme de resolution de contraintes dans un systeme d'orchestration assistee par ordinateur*, Seminaire du groupe de travail Musique, Contraintes et Interaction (AFIM), Vannes, France, janvier 2008

D. Tardieu, G. Peeters, X. Rodet. An Instrument Timbre Model For Computer Aided Orchestration. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA, New Paltz, New York, 2007.

2 Résultats scientifiques et technologiques

2.1 Indexation audio et navigation par le contenu

2.1.1 SP2-1 – Module d'extraction de descripteurs audio pour les samples

Responsable : Ircam-AnaSyn

L'objectif de cette tâche est de produire une bibliothèque calculant un ensemble exhaustif de descripteurs à partir de l'analyse des fichiers sons d'échantillons.

La bibliothèque IrcamDescriptor est fournie sous forme de code C++ multi-plateforme et a été compilée pour MacOSX, Windows et Linux. Elle fournit les 54 descripteurs suivants, pour lesquels sont indiqués le type SDIF correspondant :

	dim.	var.	SDIF type
Temporal descriptors			
AutoCorrelation	v	1	1ARR
SignalZeroCrossingRate	1	1	1ZCR
Energy descriptors			
TotalEnergy	1	1	1NRG
HarmonicEnergy	1	1	1HEN
NoiseEnergy	1	1	1NEN
EnergyEnvelope	1	1	1EEV
Spectral descriptors			
SpectralCentroid	1	6	1SCN
SpectralSpread	1	6	1SSP
SpectralSkewness	1	6	1SSK
SpectralKurtosis	1	6	1SKU
SpectralSlope	1	6	1SSL
SpectralDecrease	1	1	1SDE
SpectralRollOff	1	1	1SRO
SpectralVariation	1	3	1SVA
Harmonic descriptors			
FundamentalFrequency	1	1	1FQ0
Noisiness	1	1	1NSN
Inharmonicity	1	1	1INH
HarmonicSpectralDeviation	1	3	1HSD
HarmonicOddToEvenRatio	1	3	1HOE
HarmonicTristimulus	3	3	1HTR
HarmonicSpectralCentroid	1	6	1HCN
HarmonicSpectralSpread	1	6	1HSP
HarmonicSpectralSkewness	1	6	1HSK
HarmonicSpectralKurtosis	1	6	1HKU
HarmonicSpectralSlope	1	6	1HSL
HarmonicSpectralDecrease	1	1	1HDE
HarmonicSpectralRollOff	1	1	1HRO
HarmonicSpectralVariation	1	3	1HVA
Perceptual descriptors			
PerceptualSpectralDeviation	1	3	1PSD
PerceptualOddToEvenRatio	1	3	1POE
PerceptualTristimulus	3	3	1PTR
PerceptualSpectralCentroid	1	6	1PCN
PerceptualSpectralSpread	1	6	1PSP
PerceptualSpectralSkewness	1	6	1PSK
PerceptualSpectralKurtosis	1	6	1PKU
PerceptualSpectralSlope	1	6	1PSL
PerceptualSpectralDecrease	1	1	1PDE
PerceptualSpectralRollOff	1	1	1PRO
PerceptualSpectralVariation	1	3	1PVA

Loudness	1	1	1LDN
RelativeSpecificLoudness	v	1	1RSL
Sharpness	1	1	1SHA
Spread	1	1	1SPR
SpectralFlatness	4	1	1SFM
SpectralCrest	4	1	1SCM
MFCC	v	1	1MFC
DMFCC	v	1	1DMF
DDMFCC	v	1	1DDM

Figure 41 - Descripteurs calculés avec leur signature SDIF

Usage of the `ircamdescriptor` tool

Mise à part la bibiothèque, plusieurs exécutables ont été réalisés pour le test des fonctions. Dans le cas de IrcamDescriptor (ligne de commande), la syntaxe d'utilisation de l'exécutable est la suivante :

```
ircamdescriptor soundfile.aif config.txt v
```

La sortie est un fichier SDIF appelé `soundfile.aif.descr.sdif` contenant les descripteurs. L'option `v` sélectionne le mode verbose, dans lequel les noms et valeurs des descripteurs sont affichés.

Tous les paramètres d'analyse et de sélection individuelle des descripteurs sont définis dans un fichier texte de configuration et passés en deuxième argument. Les réglages de `config.txt` sont divisés en trois groupes: le premier de header `[Parameters]`, concerne les paramètres généraux d'analyse comme les longueurs de fenêtres, de pas d'incrémentation d'analyse, de paramètres d'extraction f0, etc. Les réglages de header `[StandardDescriptors]` concernent les descripteurs instantanés décrits en Figure 41. Ceux de header `[EnergyDescriptors]` concernent l'enveloppe d'énergie(`1EEV`), qui est habituellement calculée avec une résolution temporelle plus élevée que les autres descripteurs instantanés, et les modélisations temporelles associées (cf. Figure 3). La plupart des paramètres du fichier se comprennent d'eux-mêmes. Les détails des autres paramètres sont donnés dans le rapport technique WP2-M22.

2.1.2 SP2-2 – Module d'indexation automatique

2.1.2.1 Classification automatique

Ce module de classification automatique prend en entrée un fichier son et produit en sortie un nom de classe et un ensemble de probabilités pour chaque classe. L'apprentissage a été entraîné sur les 3 taxonomies, issues des expériences psychoacoustiques (cf § 2.1.5.2) qui sont rappelées ci-après :

Taxonomie	**Causale**		**Causale réduite**		**Onomatopées**	
Nbre de classes	7		5		5	
Précision de choix aléatoire	14,29%		20,00%		20,00%	
Nom des classes et nombre de fichiers	1. Electrical	66	1. Electrical	66	1. Buzzing	124
	2. Fire	64	2. Gas	135	2. Crackling	159
	3. Gas	135	3. Liquid	269	3. Rushes	154
	4. Liquid	269	4. Solid-metal	192	4. Scratching	48
	5. Solid-metal	192	5. Solid-paper	67	5. Squeaking	106
	6. Solid-paper	67				
	7. Solid-wood	103				
Nbre total de fichiers son		896		729		591

Figure 42 – Taxonomies finales

Les matrices de confusion obtenues pour les paramètres optimaux issus du processus de test d'apprentissage sont données ci après en Figure 43, Figure 44 et Figure 45. Les valeurs dans les diagonales donnent les pourcentages de reconnaissance correcte pour chaque classe.

real/found	Gas	Liquid	Solid-Metal	Solid-Paper	Solid-Wood	Fire	Electrical
Gas	**56,92**	1,54	12,31	1,54	21,54	1,54	4,61
Liquid	0	**59,38**	15,63	9,38	9,38	0	6,25
Solid-Metal	1,49	2,24	**73,88**	14,18	6,72	0	1,49
Solid-Paper	1,12	2,24	4,48	**85,82**	2,99	1,87	1,49
Solid-Wood	2,07	2,07	4,15	2,59	**76,68**	3,11	9,33
Fire	0	0	0	3,08	4,62	**78,46**	13,85
Electrical	0	1,92	0	2,88	31,73	4,81	**58,65**

Figure 43 – Matrice de confusion pour la taxonomie causale, paramètres optimaux

real/found	Gas	Liquid	Solid-Metal	Solid-Paper	Electrical
Gas	**55,38**	6,15	6,15	32,31	0
Liquid	1,48	**86,67**	6,67	5,19	0
Solid-Metal	0	4,83	**87,36**	6,69	1,12
Solid-Paper	1,04	4,15	3,63	**88,08**	3,11
Electrical	0	0	8,82	13,24	**77,94**

Figure 44 – Matrice de confusion pour la taxonomie causale réduite,paramètres optimaux

real/found	Buzzing	Crackling	Squeaking	Scratching	Rushes
Buzzing	**75,81**	4,03	11,29	2,42	6,45
Crackling	3,77	**77,99**	6,29	1,26	10,69
Squeaking	3,90	8,44	**77,27**	1,30	9,09
Scratching	8,33	6,25	10,42	**52,08**	22,92
Rushes	18,10	9,52	4,76	6,67	**60,95**

Figure 45 - Matrice de confusion pour la taxonomie d'onomatopées, paramètres optimaux

Utilisation de l'outil `ircamclassifier`

L'utilisation des outils standalone implique le calcul préalable des descripteurs avec un fichier texte de configuration approprié:

```
ircamdescriptor soundfile.aif config_onomat.txt
```

L'étape suivante consiste à passer le fichier SDIF calculé par ircamdescriptor à IrcamClassifier pour effectuer la classification:

```
ircamclassifier soundfile.aif.descr.sdif onomat.model.svm v
```

L'option `v` correspond au mode verbose, dans lequel toutes les probabilités de chaque classe sont affichées:

```
001 Buzzing............: 0.08415784846404986
002 Crackling..........: 0.5822374175372725
003 Squeaking..........: 0.01007221644053237
004 Scratching.........: 0.008384234985555832
005 Rushes.............: 0.3151482825725895
```

```
best class: *Crackling* (0.5822374175372725)
```

La Figure 46 indique les fichiers de configuration et de modèle SVM à utiliser pour chaque taxonomie.

taxonomy	IrcamDescriptor config file	IrcamClassifier model file
Causal	config_causal.txt	causal_3-10-2008.model.svm
Causal-small	config_causal_small.txt	causal_small_3-10-2008.model.svm
Onomat	config_onomat.txt	onomat_3-10-2008.model.svm

Figure 46 – Fichiers de configuration et de modèle SVM pour les trois taxonomies

2.1.2.2 Description et classification morphologiques

L'objectif de cette tâche était de caractériser automatiquement les sons en fonction de leur profil dynamique, selon les classes *stable*, *train d'impulsions*, *décroissant*, *impulsif*, *roulement* et *croissant* issus des expériences psychoacoustiques réalisées (cf §2.1.4.2).

Les arbres de classification binaires obtenus pour la classification des profils dynamiques ont été intégrés comme modèles supplémentaire dans IrcamClassifier. Ainsi, l'utilisation de IrcamDescriptor et IrcamClassifier pour la classification morphologique est la même que pour la classification statistique, en utilisant les fichiers appropriés de configuration et de modèle, montrés en Figure 47. IrcamClassifier produit la sortie suivante avec la meilleure classe :

```
best class: *Ascending* (1)
```

taxonomy	IrcamDescriptor config file	IrcamClassifier model file
dynamic profiles	config_dynmorph.txt	dynmorph_3-10-2008.model.dyn

Figure 47 – Fichiers de configuration et de modèle pour la classification morphologique de profils dynamiques

2.1.2.3 Recherche par similarité

L'objectif de cette tâche était d'élaborer une mesure de similarité entre sons à partir de leurs résultats de classification. Deux fichiers de modèles ont été fournis (causal réduit et onomatopées), contenant les données nécessaires à la transformation par ALD pour réduire la dimensionnalité des descripteurs à 2D. L'utilisation pour produire les indices d'un fichier est le même que pour la classification : extraction des descripteurs avec IrcamDescriptor, avec le même fichier de configuration et utilisation de IrcamClassifier avec le fichier de modèle approprié (cf noms en Figure 48).

taxonomy	IrcamDescriptor config file	IrcamClassifier model file
Causal-small	config_similarity.txt	causal_small_3-10-2008.similarity.svm
Onomat	config_similarity.txt	onomat_3-10-2008.similarity.svm

Figure 48 - Fichiers de configuration et de modèle pour la similarité dans un espace de dimensionnalité réduite par ALD

2.1.3 SP2-3 – Module d'extraction de descripteurs audio pour les boucles

Responsable : Ircam-AnaSyn

L'objectif de cette tâche était de caractériser les échantillons de boucles selon deux aspects principaux : le rythme (tempo, structure métrique) et la tonalité. Les bibliothèques C++ préexistantes IrcamBeat et IrcamKeyMode ont été adaptées à cet effet et rendues

disponibles sur plateformes MacOSX et Windows. Les détails de leur utilisation sont décrits au § 1.3.4.

2.1.4 SP2-4 – Description des sons environnementaux

2.1.4.1 D-SP2-4-1 – Validation de descripteurs audio pour les sons environnementaux

Responsable : Ircam-PDS

L'objectif de cette sous-tâche était de proposer une organisation des sons environnementaux à partir de la base de données de référence fournir par Univers Sons. Les expériences réalisées sont décrites au § 1.3.6.1. Les données de l'expérience 1, traitées par une analyse de cluster ont confirmé la première partie de l'hypothèse (cf.Figure 16) et permis de proposer une organisation des sons à **2 niveaux** de ce corpus spécifique:

- un **niveau discret** possédant trois classes d'événement :

. **Impact** : courtes impulsions ou séries d'impulsions
. **Moteur** : mélange harmonique/bruit
. **Pseudo-Instrument** : production similaire à celle d'un instrument de musique

- un **niveau continu** permettant de définir un espace multidimensionnel de timbre commun aux sons de chaque classe, avec

. un attribut commun à toutes les classes : la ***brillance*** (avec, pour la classe Moteur, la prise en compte des parties harmonique et bruitée)

. des attributs propres, liés aux spécificités acoustiques de chaque classe : la ***netteté*** pour les sons d'impact, l'***émergence harmonique*** pour les sons de moteur, et la ***rugosité*** et la ***richesse spectrale*** pour les sons pseudo-instrumentaux (se reporter à [Minard08a] pour plus de détails sur la définition et la formulation de ces différents descripteurs).

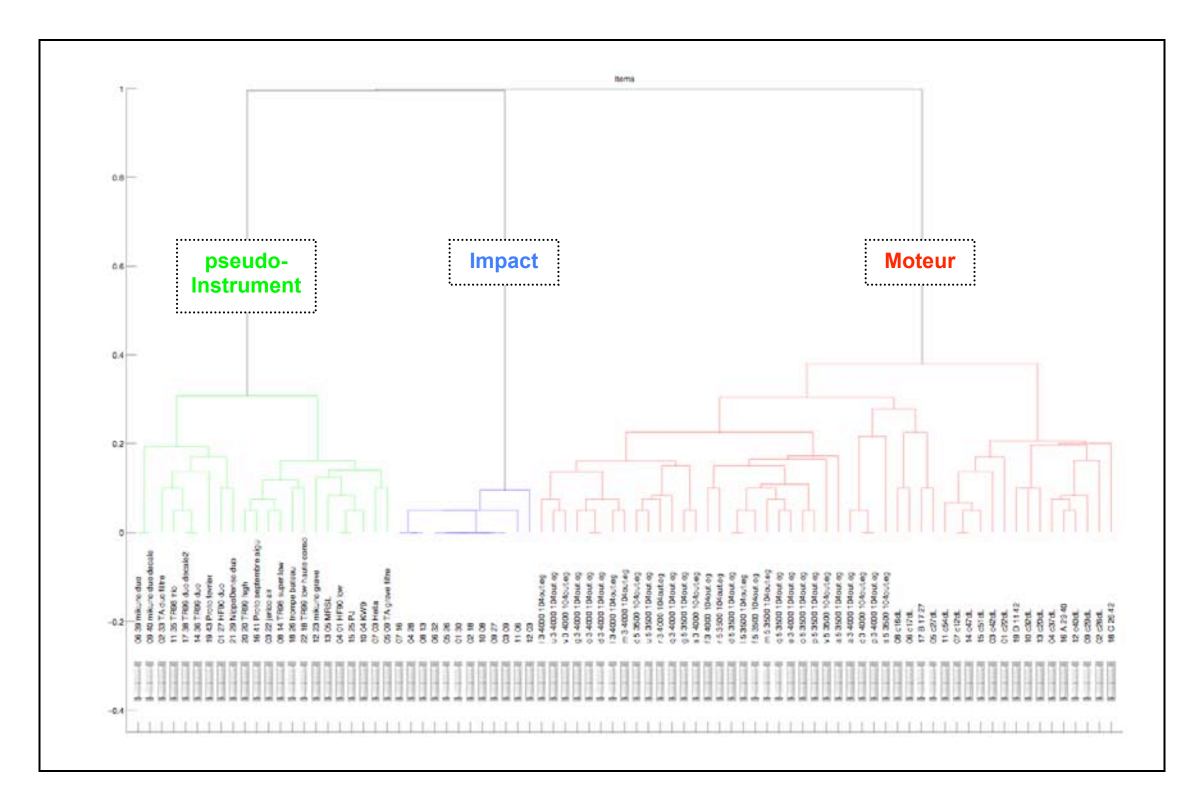


Figure 49 - Méta-analyse– résultats Expérience 1

Les données de la seconde expérience ont permis de valider perceptivement un corpus élargi définissant le modèle à 3 classes précédemment établi : l'analyse par moyenne et écart-type sur l'ensemble des participants, du nombre de sons placés dans chaque classe permet d'aboutir à une base de **124 sons** : 47 Impact, 50 Moteur, 27 pseudo-Instrument, et donc 26 sons rebutés (cf.Figure 50).

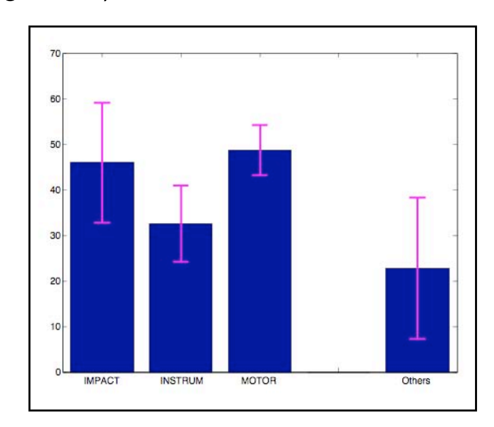


Figure 50 - Méta-analyse– résultats Expérience 2

2.1.4.2 D-SP2-4-2 – Descripteurs audio de type morphologique pour les sons environnementaux

Responsable : Ircam-PDS

L'objectif de cette tâche était de caractériser des description de sons environnementaux en termes de profils morphologiques. Les expériences effectuées sont décrites au §1.3.6.2. Les résultats concernant les profils mélodiques se sont avérés non concluants. En effet, la première expérience (classification libre + tracé) n'a pas permis d'obtenir des classes mélodiques stables. L'une des causes de cette impasse est certainement à chercher dans

les sons eux-mêmes (par nature environnementaux donc ne possédant pas de hauteurs bien définies), d'où la difficulté des participants à classer les sons selon des critères mélodiques.

Les résultats concernant les profils dynamiques sont plus tangibles:

- Les données de l'expérience 1 sont traités par une analyse de cluster (méthode 'average') qui permet d'extraire 6 classes dynamiques pour les sons environnementaux : **stable, train d'impulsions, décroissant, impulsif, roulement** et **croissant** (voir Figure 51)

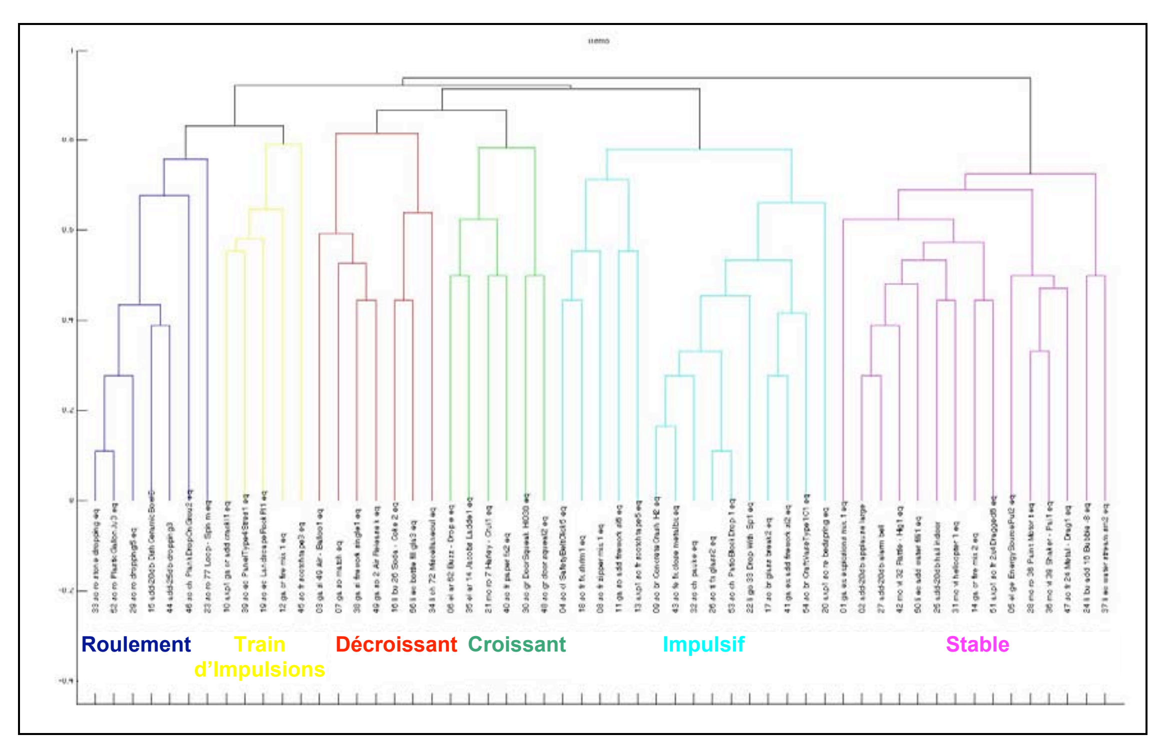


Figure 51 - Descripteurs morphologiques / catégorisation libre – résultats (analyse de cluster)

- L'expérience 2 confirme, en premier lieu, la configuration de classification 'moyenne' obtenue à l'étape précédente : le moyenne des jugements varie entre 6.5 et 8.5 (écarts-type entre 1 et 2.5). Elle permet également de formaliser, pour chacune de ces classes des éléments de symbolisme (voir Figure 53) basés sur une analyse des tracés individuels (voir Figure 52)

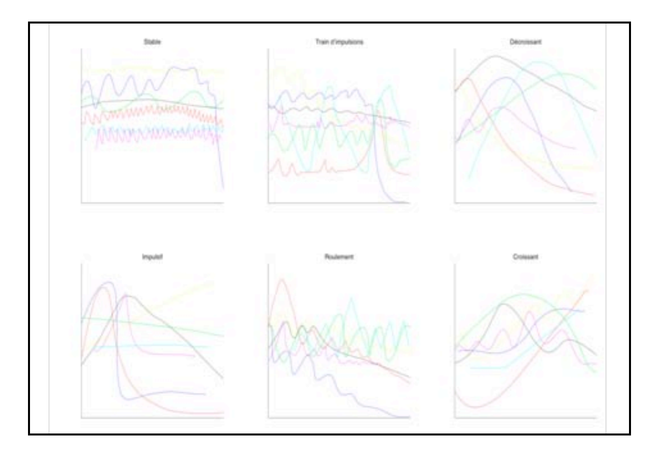


Figure 52 - Descripteurs morphologiques / classes ‘moyennes’ - résultats individuels

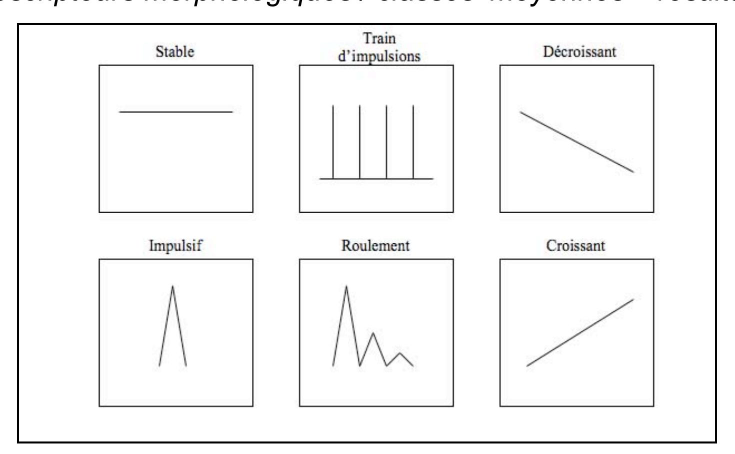


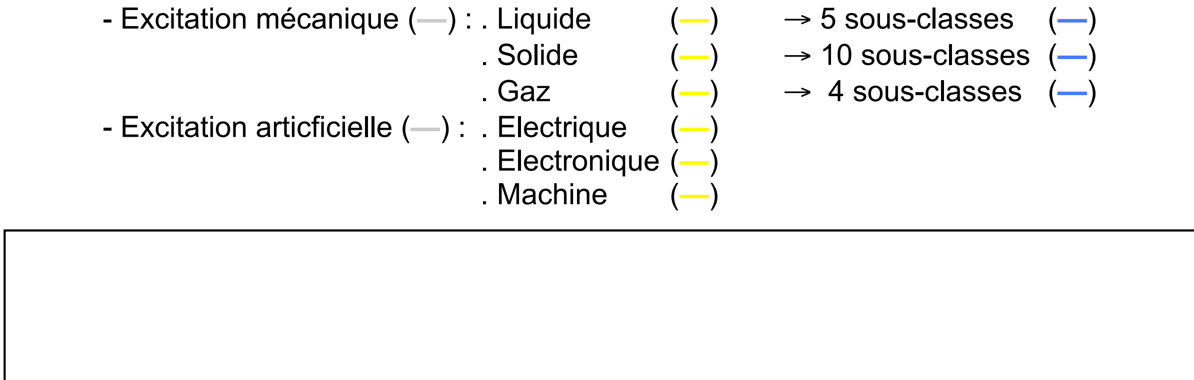
Figure 53 - Descripteurs morphologiques / classes ‘moyennes’ - symboles

2.1.5 SP2-5 – Classification des sons environnementaux

2.1.5.1 D-SP2-5-1 – Définition d’un corpus représentatif et sélection des classes pertinentes

Responsable : Ircam-PDS

L’objectif de cette tâche était de produire, à partir d’un ensemble de sons fournis par Univers Sons, à une corpus de sons de référence pour tous les travaux de description et d’analyse automatique. Cette tâche a permis d’aboutir à un corpus structuré et utile pour la suite du travail **("**SOR_database_structured") composé de **402 sons** déployés sur **3 niveaux de classification** (voir Figure 54) :



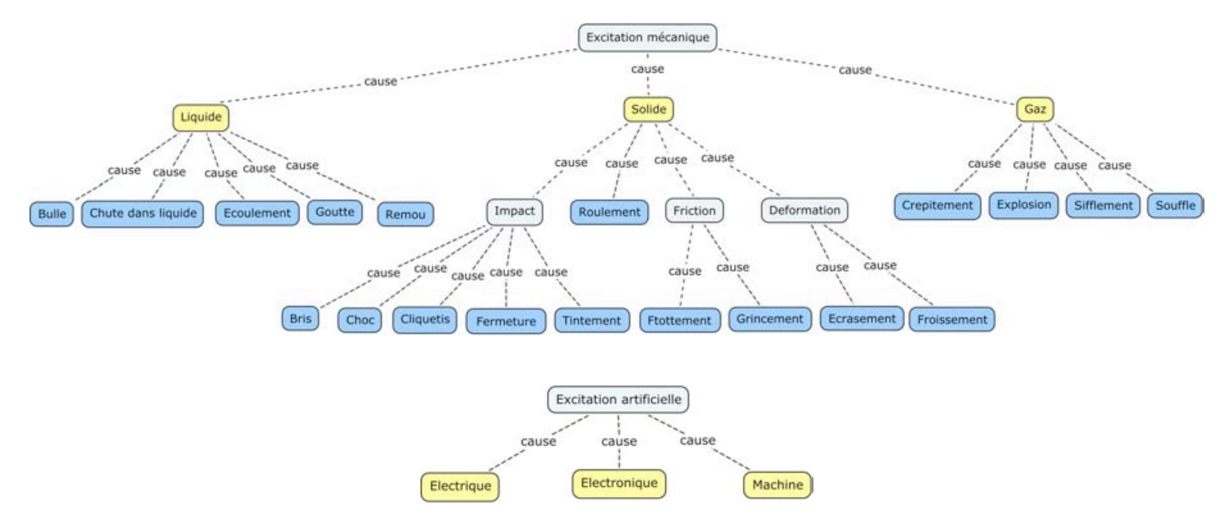


Figure 54 - Corpus sonore représentatif (Univers Sons)

Taxonomie d'usage vs. structure interne du niveau /Evènementiel/

2.1.5.2 D-SP2-5-2 – Organisation des classes pertinentes

Responsable : Ircam-PDS

L'objectif global de cette tâche était de fournir des éléments d'intégration pour le SP2-2, sous la forme d'un corpus classifié et validé perceptivement, afin de tester les méthodes de classification automatique. Les résultats issus des expériences réalisés permettent d'aboutir aux conclusions suivantes :

- d'une part, le modèle de taxonomie causale issue des travaux de [Gaver93] et [Houix07b] semble bien adapté à la base de données de travail, ce qui renforce sa validité et sa représentativité ;

- d'autre part, l'objectif fixé peut être atteint : générer un corpus de sons classifié et validé d'un point de vue perceptif afin d'alimenter les procédures de classification automatique développée dans SP2-2. Pour cela, une sélection des sons en fonction de la répartition des réponses est effectuée au 1er niveau de description et aboutit dans un premier temps à un corpus de 186 sons (30– Liquide, 83– Solide, 38– Gaz, 7– Electrique, 13– Electronique, 15– Machine). Dans un second temps, pour des raisons de nombre d'éléments nécessaires à la validation des classificateurs automatiques, un ré-équilibrage des classes est effectué et aboutit *in fine* à une base de travail livrée à SP2-2 de **245 sons** (30– Liquide, 83– Solide, 38– Gaz, 35– Electrique, 31– Electronique, 28– Machine).

2.1.6 SP2-6 – Caractérisation sémantique des sons environnementaux

2.1.6.1 D-SP2-6-1 – Description sémantique des sons

Responsable : Ircam-PDS

Les résultats de cette tâche sont décrits au §1.3.8.1.

2.1.7 SP2-7 – Recherche, navigation et composition interactive fondée sur des bases de sons

Les résultats réalisés par le projet *SampleOrchestrator*, détaillés dans les sections suivantes, s'organisent autour du développement de CataRT comme application prototype pour expérimenter des paradigmes de recherche et création sonore par navigation dans des grandes collections sonores, ainsi que des modules, bibliothèques de fonctions, et sous-

systèmes, qui ont été développés et testés dans CataRT, mais qui sont intégrables et utilisables dans d'autres applications. Les points suivants ont été développés :

- Librarie *DistLib* (1.3.9.2) contenant des algorithmes de recherche efficaces dans des grandes collections de données, notamment par arbres de recherche binaires (k*D-Trees*)
- Librarie *VisLib* pour le support de la visualisation de données multidimensionnelles (1.3.9.2) en utilisant des méthodes de réduction de dimensionnalité tel que multi-dimensional scaling (MDS) et principal component analysis (PCA) avec l'intégration de poids par dimension.
- Avancement de l'architecture et des fonctionnalités de CataRT comme modèle d'une application de navigation dans des grandes bases sonores, et de sa documentation (1.3.9.4)
- Nouvelle approches à la visualisation pour la recherche de sons et la navigation dans des grandes collections sonores (1.3.9.5)
- Version *standalone* simplifié de CataRT, servant de démonstrateur
- Modélisation de la continuation de boucles rythmiques (1.3.9.6, voir aussi SP3-4 – Variation interactive de boucles rythmiques) : ces recherches ont été effectués au sein de l'application CataRT et ont résulté dans des parties de la librairie *DistLib* et des nouvelles fonctionnalités de CataRT.

2.1.7.1 D-SP2-7-3 – Application et modules finaux de navigation et synthèse par corpus

Responsable : Ircam-IMTR

2.1.7.2 Application démonstrateur: SOREX—SampleOrchestrator Sound Explorer

La version standalone simplifiée SOREX (SOR–SoundExplorer), sert de démonstrateur et intègre les nouvelles fonctionnalités dans CataRT , développées en libraries et autres modules. Ce démonstrateur tourne sous Max/MSP ou comme application indépendante. Il met en œuvre la scalabilité de la recherche et synthèse pour des grands corpus par sélection d'unités optimisé utilisant un index multidimensionnel *kD-Tree*, la lecture des sons directement du disque, l'optimisation et accélération de la visualisation pour la recherche de sons par *zoom & pan* et l'intégration des descripteurs catégorielles (classes, taxonomie) et morphologiques dans la représentation avec un nouveau mode de visualisation par *pavage (tile)*

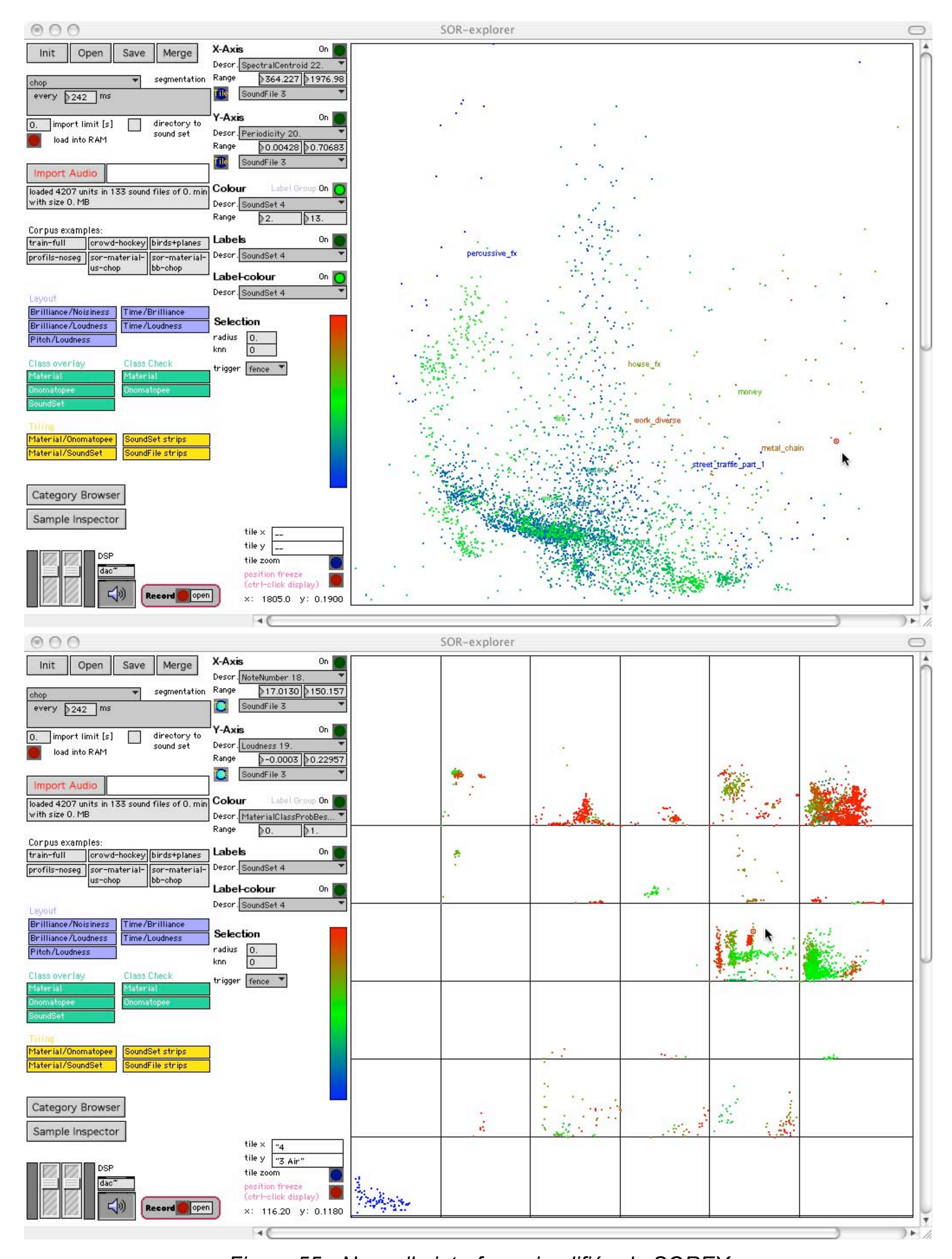


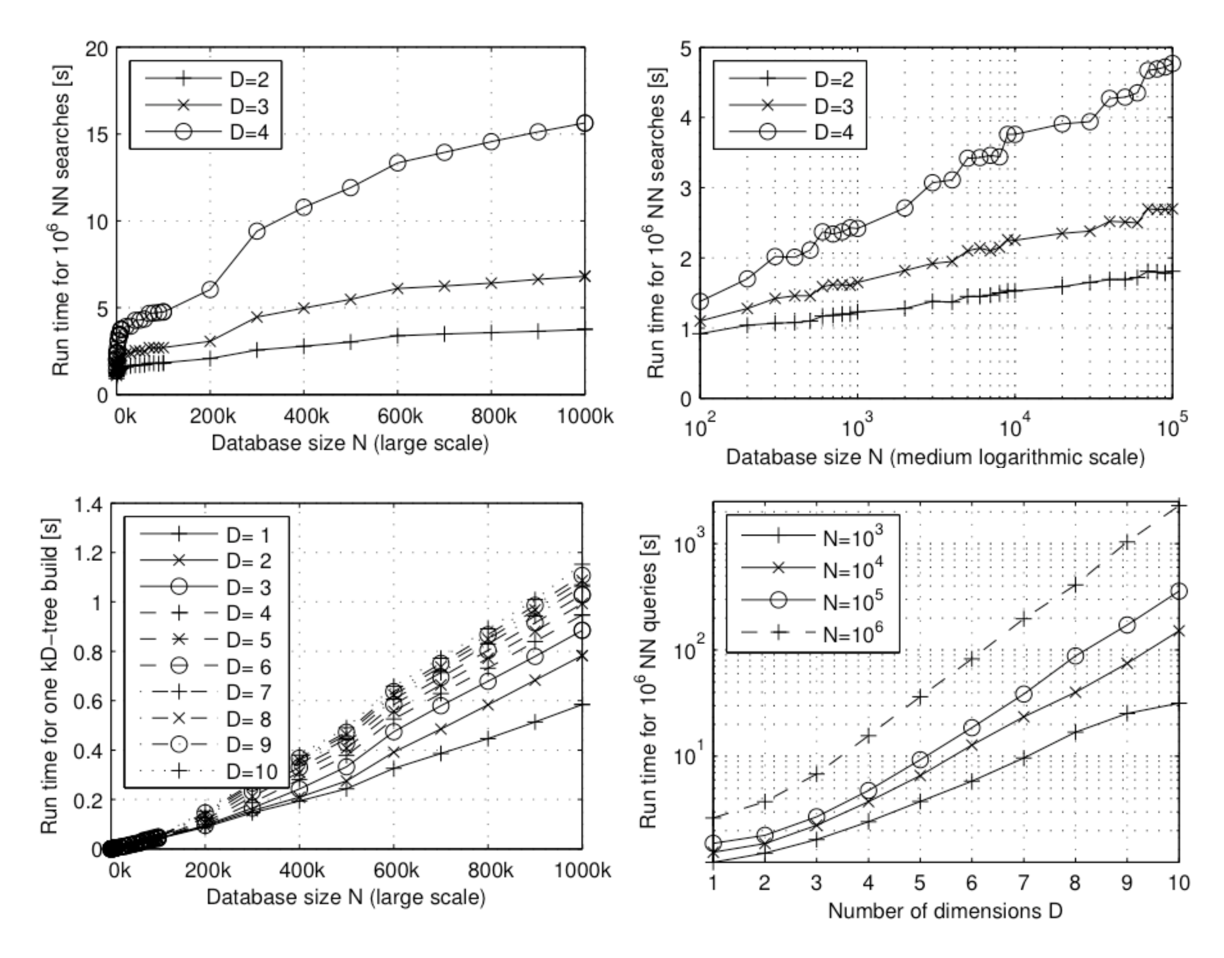
Figure 55 - Nouvelle interface simplifiée de SOREX

2.1.7.3 Librarie DistLib: Algorithmes de recherche efficaces dans des grandes collections de données multidimensionnelles

La librairie *DistLib* intègre les algorithmes de recherche efficaces dans des grandes collections de données multidimensionnelles par arbre kD-Tree. Elle est écrite en C portable et disponible pour Mac (universal binary) et Windows.

Les mesures des performances de l'algorithme de recherche dans des données multidimensionnelles par arbre binaire (k*D-Tree*) sont présentées comme le temps de CPU en millisecondes que le processus consomme pour la recherche de *r* vecteurs de dimension *d* tirées au hasard dans des données de taille *n* (c'est à dire une matrice (*n, d*) de *n* lignes de vecteurs éléments à dimension *d*, également tirées au hasard pour chaque nouvelle valeur de *n*.

Les diagrammes de mesures en haut montrent en abscisse le temps logarithmique pour la recherche de r = 10^6 points par l'algorithme kD-Tree en secondes pour différents tailles n. Le temps linéaire de construction de l'arbre de recherche est affiché en bas à gauche. La figure en bas à droite montre l'influence exponentielle de la dimensionnalité d. Par contre, le temps de recherche pour un point le plus long pour n = 10^6 et d = 10 est seulement de 2.2 ms sur un processeur 2.53 GHz Intel Core 2. Ces mesures ont été prises dans Max/MSP en retirant le temps mesuré sans que l'algorithme de recherche soit appelé. Ainsi, le overhead du système de mesure n'est biaise pas les résultats.



2.1.7.4 Librarie VisLib *pour la Visualisation de données multidimensionnelles*

La librairie *VisLib* intègre les algorithmes pour la visualisation de grandes collections de données multidimensionnelles par *Hybrid MDS* et le modèle masse-ressort sous-jacent. Elle est écrite en C portable et disponible pour Mac (universal binary) et Windows.

Le modèle masse-ressort simule un système physique de masses en 2D liés par des liens exerçant des forces et accélérations sur ces masses (sans inertie pour éviter des oscillations). Si les longueurs nominaux des liens correspondent aux distances haut-dimensionnelles, l'état stable résultant de la simulation correspond à une disposition 2D qui représente bien les contraintes de voisinage des points.

Aussi, pour la disposition des points, l'effet de répulsion entre les points peut mieux distribuer des groupes de points superposés, afin de pouvoir voir tous les sons, comme montré dans la figure suivante :



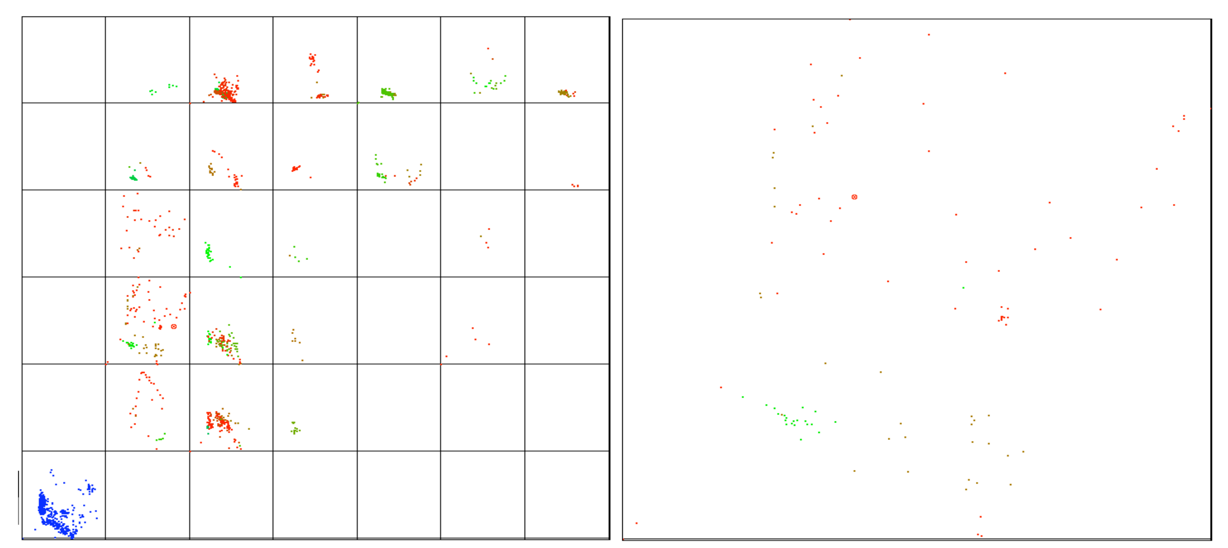
2.1.7.5 Développement de l'architecture et des fonctionnalités de CataRT

L'amélioration de l'architecture de CataRT permettait une meilleure modularité, afin d'intégrer plus simplement des nouvelles fonctionnalités. Le système modulaire est distribué en forme de patch Max/MSP avec extensions FTM&Co. sous License GPL. Notamment le nouveau modèle du système graphique de CataRT qui est organisé dans 3 couches de coordonées : les descripteurs, les coordonnées de présentation, et les coordonnées de sortie, permet une définition plus aisé et flexible des différents transformations entre les couches, qui réalisent par exemple la réduction de dimensionnalité, la normalisation, la juxtaposition de sous-espaces, ou l'agrandissement (zoom).

D'autres extensions de l'interface interactive de navigation entre échantillons et de synthèse par corpus sont l'intégration des classes (description, recherche et navigation), le contrôle de la sélection par un son cible (fichier son, MIDI), et une documentation étendue (tutoriaux, mode d'emploi).

2.1.7.6 Visualisation

Le mode *tile* permet la comparaison de deux descripteurs (axes x et y) selon des appartenances à une ou deux catégories (classes issu de la classification automatique ou groupes de sons définis par l'utilisateur). La figure gauche montre la vue d'ensemble en mode mosaique (tile). En appuyant sur la touche T, l'utilisateur peut sauter entre la vue d'un *tile* agrandi et la vue d'ensemble, comme illustré dans la figure droite montrant la vue agrandi d'une sous-catégorie.



La fenêtre du *Category Browser* permet le filtrage des unités selon des catégories. Pour la sélection d'une classe ou d'une combinaison de classes et autres catégories, le mode *Solo* élimine toutes sauf les catégories choisis de la sélection dans l'espace de navigation. Cependant, les unités exclus restent affichés en gris pour garder le contexte et pour montrer les unités présentes à une position de descripteurs donnée, car il se peut qu'un son hors des catégories choisis soit quand même intéressant car pourrait remplir mieux certains besoins en termes de descripteurs.

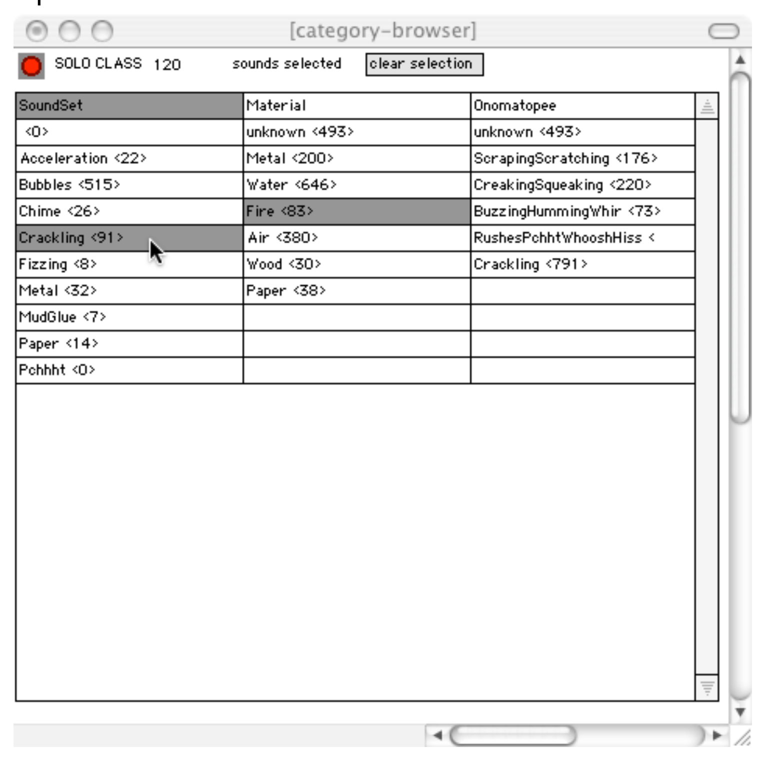


Figure 56 - Dialogue de choix de catégories et classes à filtrer

2.2 Analyse et traitement audio

2.2.1 SP3-1 – Moteur d'analyse et de transformations sonores temps réel

Responsable : Ircam-AnaSyn

2.2.1.2 Module de base

La réalisation a consisté en une livraison de la bibliothèque SuperVP (2.97.8) pour les plateformes MacOSX et Windows ayant fait l'objet d'un ensemble d'optimisations (cf § 1.4.1.2), qui ont permis d'augmenter la performance obtenue sous Windows par un facteur 6-10. Les modifications donnent des améliorations de la performance suivante :

- Changement de la bibliothèque FFT : facteur (1.5-2)
- Utilisation des opérateurs SIMD (SSE/SSE2) pour les fonctions mathématiques : facteur (1.3-1.5)
- Remplacement du compilateur MSVC (2008/2005) par compilateur Intel Version 10 : facteur (1.3-1.8)
- Synthese a partir de pre-analyse : facteur 10

La première version de la dll pour Windows a été délivrée a Univers Son à l'échéance M18.

2.2.1.3 Transposition en domaine spectral

La transposition en domaine spectral est complètement fonctionnelle tel qu'elle peut remplacer la transposition en domaine temporelle à partir de la version 2.96.32 de SuperVP. La transposition en domaine spectral peut être effectué à partir de sons preanalyse à partir de la version 2.97.8 de SuperVP.

2.2.1.4 Traitement avec taille de fenêtre variable

Les traitements et les analyses avec taille de fenêtre variable ont été intégrés dans SuperVP à partir de la version 2.94.18 . Cela concerne toutes les opérations possibles dans SuperVP.

2.2.1.5 Amélioration/qualification du F0 pour le contrôle

Les améliorations apportées à l'algorithme d'estimation et qualification des fréquences fondamentales ont été intégrées dans le moteur de calcul SuperVP (2.94.13) et elles sont alors désormais accessibles dans l'application *AudioSculpt* de l'IRCAM (2.9.2) où elles permettent à l'utilisateur d'adapter les paramètres de l'estimation de la F0 en choisissant l'instrument analysé et en plus par manipulation d'un curseur glissant de supprimer les parties de la fréquence fondamentale peu fiable.

Les nouveaux préréglages par instruments ont suscité l'intérêt de la société *MakeMusic* aux États-Unis qui a acheté une licence de la bibliothèque permettant l'utilisation des préréglages et aussi des nouveaux moyens de qualification de la périodicité permettant d'améliorer la segmentation des sons en notes.

2.2.1.6 Réorganisation des modules d'analyse

La réorganisation des modules d'analyse permet l'estimation de plusieurs descripteurs spectraux à partir d'une seule FFT. Elle a été finalisée dans la version 2.93.4 de SuperVP. La resynthèse à partir d'analyse permettant les transformations dans le module de synthèse a été finalisée dans la version 2.97.8 de SuperVP

2.2.1.7 Amélioration du remixage des composantes

La livraison finale consiste dans en une publication dans une revue [Zivanovic08a] et un nouveau mode de remixage des composantes qui permet la création du résiduel pour chaque pic spectral en temps réel dans le vocodeur de phase. La fonctionnalité est accessible à partir de la version 2.96.6 de SuperVP.

2.2.1.8 Détection, préservation et synthèse de modulations

Les fonctionnalités développées ont été implémenté en Matlab. La version finale permet la séparation de la modulation fréquentielle et du geste lent (mélodie) de la fréquence fondamentale d'un son et la suppression de toutes les modulations induites par le vibrato. La segmentation doit être fait par l'utilisateur.

2.2.2 SP3-2 – Moteur de synthèse granulaire temps réel

Responsable : Ircam-IMTR

Le moteur de synthèse granulaire à été réalisé, sous forme d'une bibliothèque C sous le nom *ZsaZsa*. La bibliothèque a été intégrée dans Max/MSP et dans une application prototype réalisé par Univers Sons.

Le moteur a été intégré dans plusieurs modules prototypes réalisés par l'Ircam et par Univers Sons:

- Module Max/MSP de synthèse granulaire *asynchrone*
- Module Max/MSP de synthèse granulaire *synchrone*
- Module Max/MSP de synthèse granulaire basé sur des descripteurs (voir SP3-3)
- Module Max/MSP de transformation de boucles rythmiques (voir SP3-4)
- Prototype de l'échantillonneur (SP4)

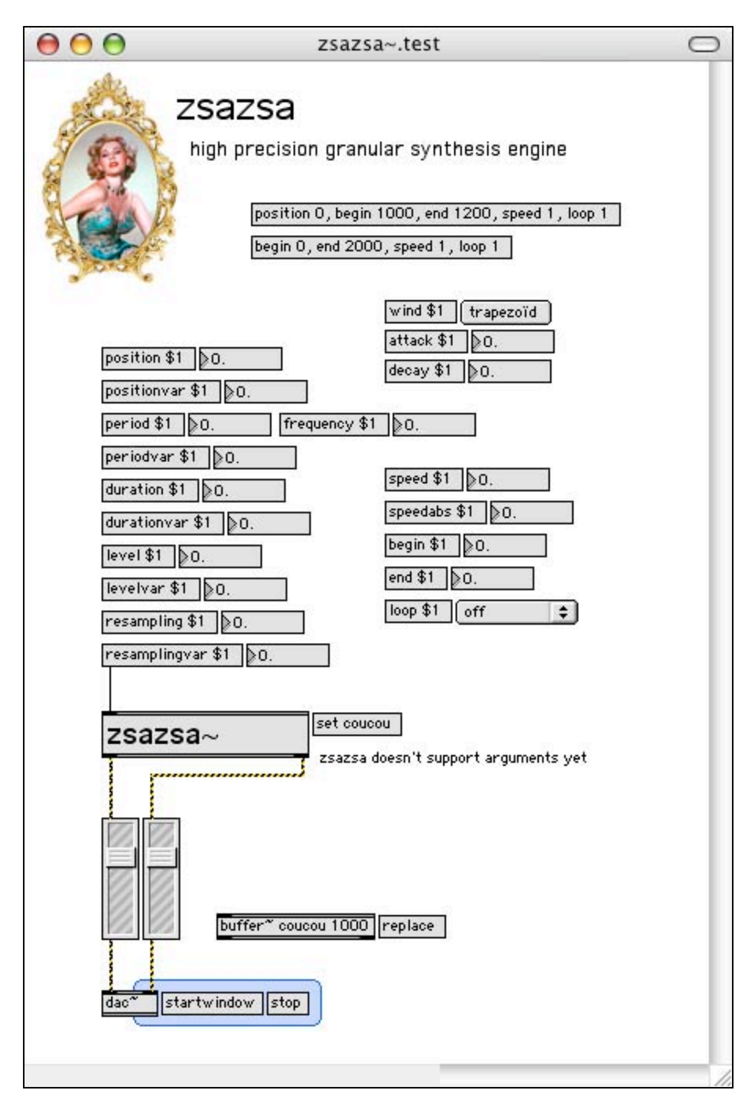


Figure 57 - Intégration dans environnement Max/MSP 4.6
d'un module de synthèse granulaire basé sur le moteur ZsaZsa.

2.2.3 SP3-3 – Synthèse granulaire basée sur les descripteurs sonores

Responsable : Ircam-IMTR

L'application finale intègre les composantes suivantes :

- Le moteur de synthèse granulaire *ZsaZsa* (SP3-2)
- Le module de sélection d'unité *KdTree* (SP2-7)
- Le conteneur de sons décrits et annotés *MuBu* (SP3-5)

De plus, les exemples de synthèse granulaire basée sur les descripteurs pour Max/MSP utilisent le module logiciel *ircamdescriptor* (SP2-1) pour l'extraction de descripteurs des fichier son.

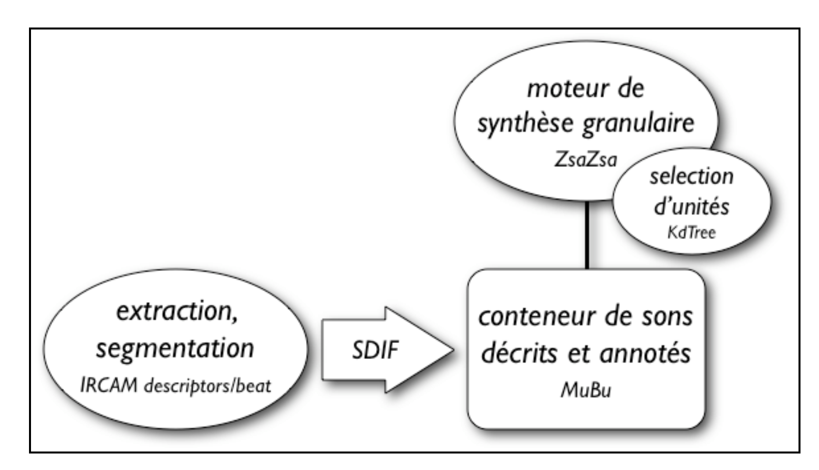


Figure 58 - Ensemble de modules pour la synthèse granulaire basée sur les descripteurs sonores.

Pour la réalisation de l'application dans Max/MSP les bibliothèques *ZsaZsa* et *KdTree*, ont été intégrées dans un module Max/MSP **sor.descr~** qui accède aux fichiers audio et descripteurs audio chargé dans le module Max/MSP **sor.mubu** (voir SP3-5 plus bas).

Deux modes de synthèse ont été réalisés dans cette application :

- Synthèse contrôlée par des valeurs cibles de descripteurs
- Synthèse croisée approximant les propriétés d'un fichier son par des grains d'un autre (« Musaicing »)

Le premier mode consiste à piloter la synthèse granulaire d'un son source par les descripteurs sonores préalablement extraits.

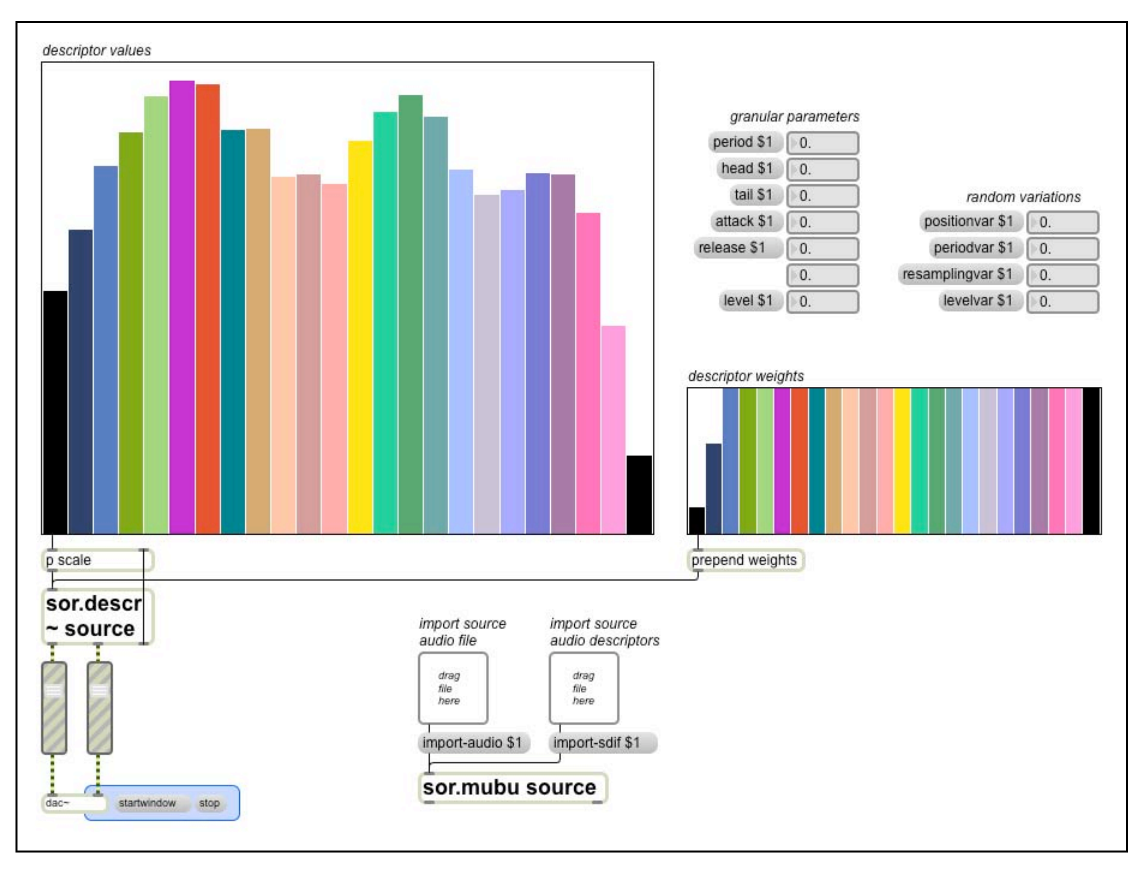


Figure 59 - Application Max/MSP 5 de synthèse granulaire basée sur les descripteurs (contrôlée par des valeurs de descripteurs).

Pour le deuxième mode, les valeurs de descripteurs et leur évolution temporelle sont déterminées par l'analyse d'un son cible selon le même ensemble de descripteurs que la source. Ce deuxième mode permet aussi la réalisation d'une forme de « morphing » (SP3-5) en jouant sur les niveaux de la synthèse granulaire du son cible et le résultat de la synthèse des grains du son source sélectionné en approximant le timbre du son cible avec des grains sélectionnés du fichier source. Dans cette application, l'utilisation des descripteurs du timbre (e.g. *MFCC*, coefficient *MEL*) s'impose.

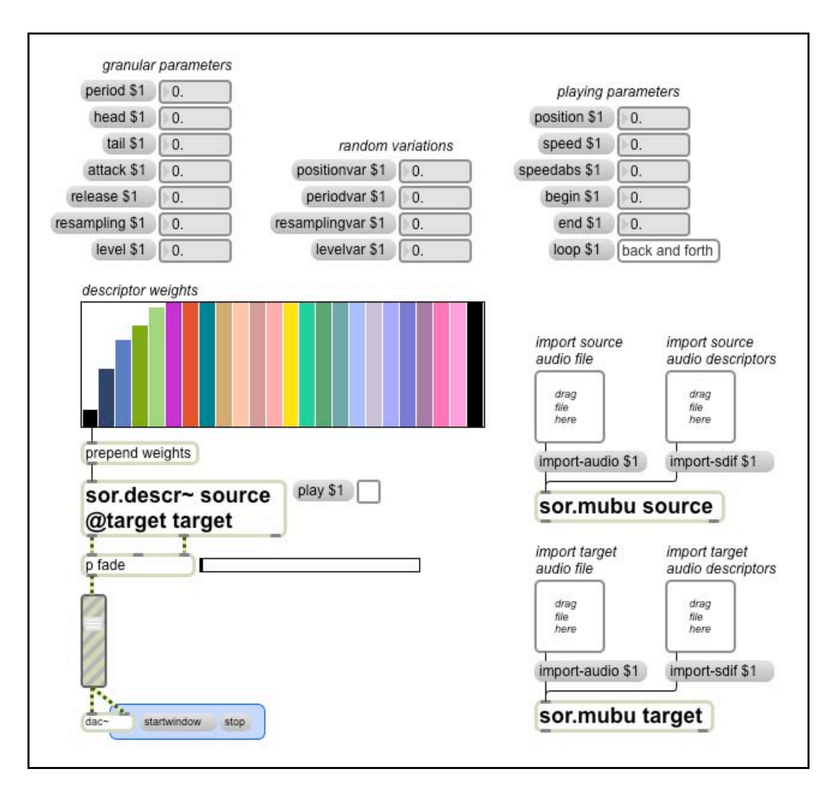


Figure 60 - Application Max/MSP 5 de synthèse granulaire basée sur les descripteurs en mode « Musaicing ».

2.2.4 SP3-4 – Variation interactive de boucles rythmiques

Responsable : Ircam-IMTR

L'application réalisée intègre deux modes de fonctionnement :

- Recomposition temporelle des segments correspondant aux pulsations
- Remplacement des segments correspondant aux pulsations dans un fichier son donné par un autre basé sur la ressemblance (selon un ensemble de descripteurs choisi)

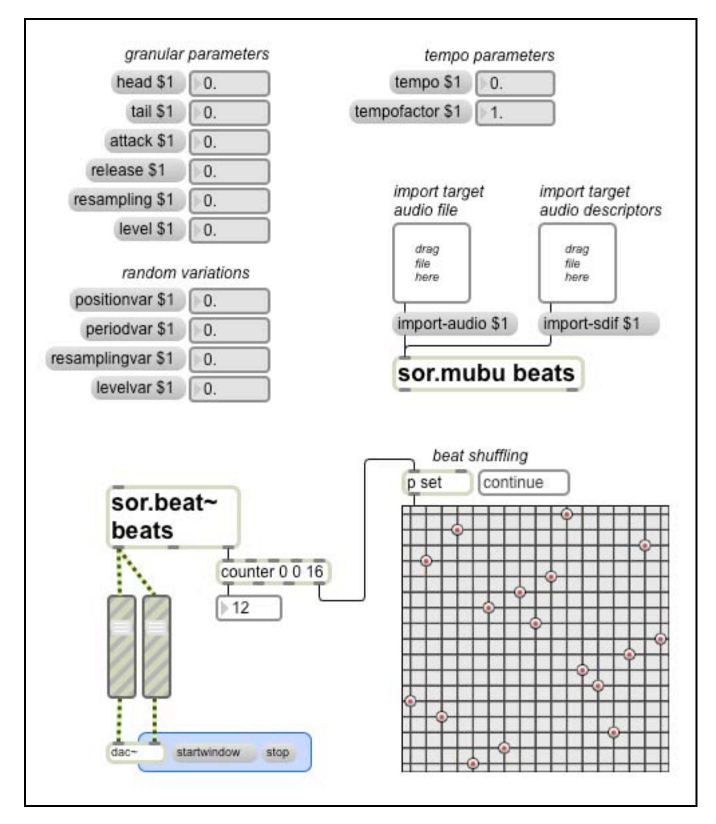


Figure 61 - Application Max/MSP 5 de transformation interactive de boucles rythmiques.

L'application Max/MSP réalisée s'appuie sur le module **`sor.beat~`** développé dans le cadre de ce sous-projet qui intègre le moteur *ZsaZsa* en mode de synthèse concaténative. Son fonctionnement nécessite un fichier audio et un fichier SDIF de marqueurs (e.g. créée avec l'outil *IRCAMbeat* développé en SP2-3) chargés dans le module **`sor.mubu`**.

2.2.5 SP3-5 – Bibliothèque MuBu

2.2.5.1 Bibliothèque MuBu

Le conteneur *MuBu* (pour « multi-buffer ») est le composant central et en commun des applications réalisées dans le cadre de SP3.

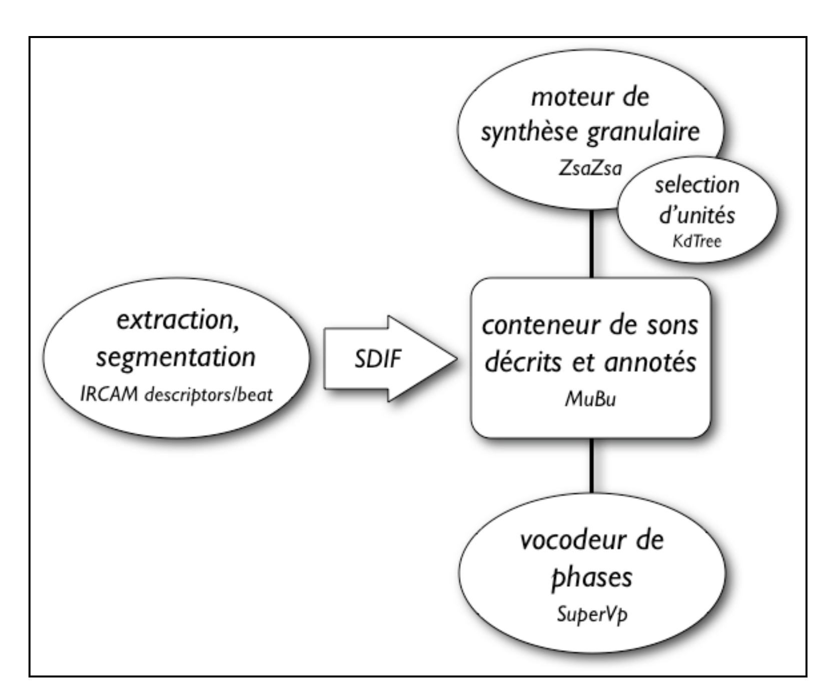


Figure 62 - Architecture de l'ensemble de composants utilisés pour la réalisation des applications de SP3 autour du conteneur MuBu.

Le conteneur est développé sous forme d'une bibliothèque C++ basée sur des modèles génériques (*« templates C++ »*). Une première version de la bibliothèque *MuBu* était disponible depuis mai 2008 avec une documentation de l'API. Une deuxième version a été finalisée en octobre 2008 pour le délivrable D-SP3-5-2.

La bibliothèque a été intégrée de manière expérimentale dans un plugin audio (*VST* et *AudioUnit*) ainsi que dans Max/MSP (en tant que module **`sor.mubu`**) pour réaliser les applications des délivrables D-SP3-3-2 et D-SP3-3-3. Le module est aussi utilisé dans un exemple d'intégration Max/MSP 5 du moteur de synthèse granulaire pour D-SP3-4-3.

La version de la bibliothèque finalisée pour le délivrable D-SP3-5-2 intègre la représentation de données en mémoire vive et les fonctionnalités suivantes :

- Représentation multipiste alignée des fichiers son, des descriptions sonores et d'autres données temporelles en mémoire vive
  - Échantillons sonores (mono ou multipistes)
  - Représentations spectrales du son
  - Descripteurs sonores et segmentations
  - Représentations simplifiées de partitions (en tant que description d'une interprétation enregistrée)
  - Données de captation du geste et du mouvement
  - Résultats d'alignement entre sons, partition et geste/mouvement
- Représentation unifiée de données de temporalité et de dimensionnalité inhomogène
  - Données échantillonnées scalaires ou multidimensionnelles
  - Données scalaires ou multidimensionnelles avec un tampon temporel
  - Segments temporels avec temps de référence, de début et de fin associant de données symboliques, scalaires ou multidimensionnelles
- Méthodes d'importation d'un maximum de format de fichiers pertinents

  - Formats de fichiers son (basée sur la bibliothèque Juce[5])
  - SDIF (*Sound Descriptor Interchange Format*)
  - MIDI standard files
  - Formats XML d'analyses sonores et formats texte
- Classe d'accès pour la création et gestion d'un conteneur multipiste
  - Création des pistes
  - Paramétrage et arrangement de pistes
  - Gestion des accès multiples et concurrents aux données
- Classe d'accès pour l'interaction avec une piste d'un containeur
  - Gestion de connexions persistantes à une piste spécifiée par index, nom ou type
  - Ensemble de méthodes d'accès aux données suivant différents paradigmes
  - Gestion des accès multiples et concurrents aux données

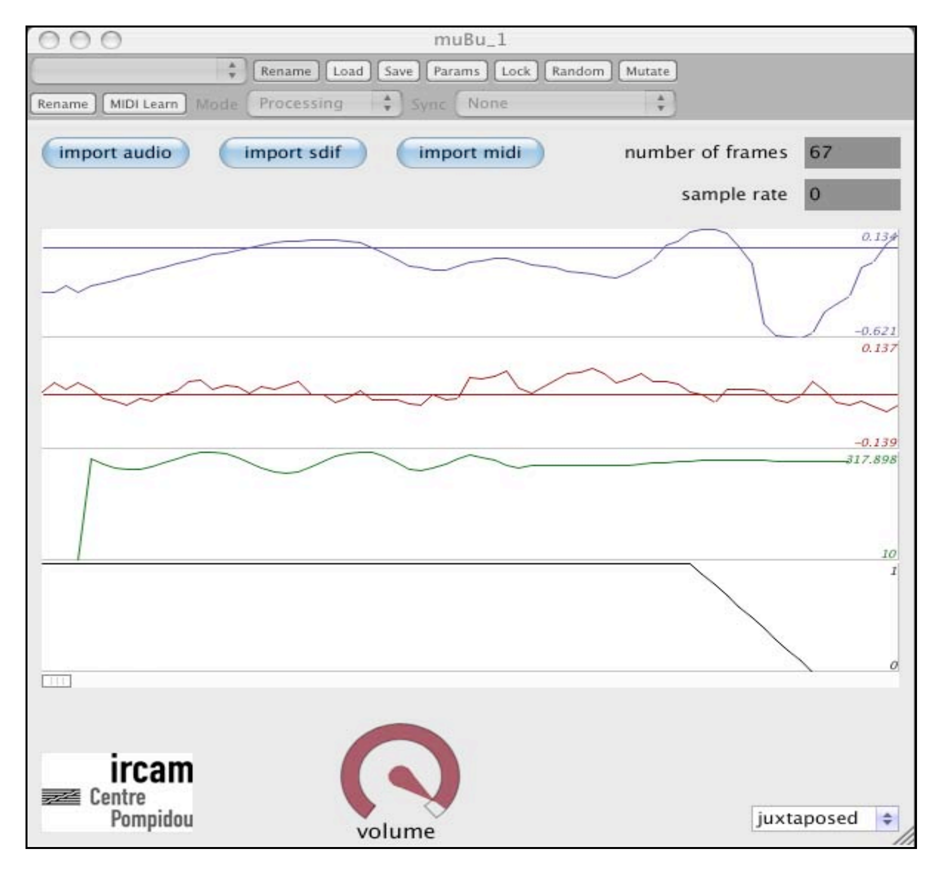


Figure 63 - Prototype d'interface intégrant la première version de la bibliothèque MuBu dans un plugin AudioUnit

La bibliothèque est entièrement basée sur des bibliothèques multi-plateformes de développement C++ (*STL*, *Boost*, *MatMTL*, *Easdif* et *Juce*) envisageant l'utilisation sur Mac OS X, Windows et Linux. Elle a été testée sous Mac OS X pendant l'élaboration et démonstration des applications du SP3 ainsi que sous Mac OS X et Windows dans d'autres prototypes d'applications qui n'ont pas été diffusés.

[5] http://www.rawmaterialsoftware.com/juce/

2.2.5.2 Morphing sonore et transformations temporelles interactives

Responsable : Ircam-IMTR

L'application permet la transformation et l'hybridation (« morphing ») de deux fichiers son chargés en mémoire vive prenant en compte les aspects suivants :

- La temporalité (rythme, articulation, phrasé)
- La hauteur (décomposée en hauteur *mélodique* et modulation)
- Le timbre (enveloppe spectrale)

L'application transforme en parallèle deux sons en fonction d'une description et d'une annotation utilisant le moteur d'analyse et de transformations sonores temps réel *SuperVP* (SP3-1). Les sons (fichiers audio) et leurs descriptions (fichiers SDIF) sont chargés en mémoire utilisant la bibliothèque *Mubu*.

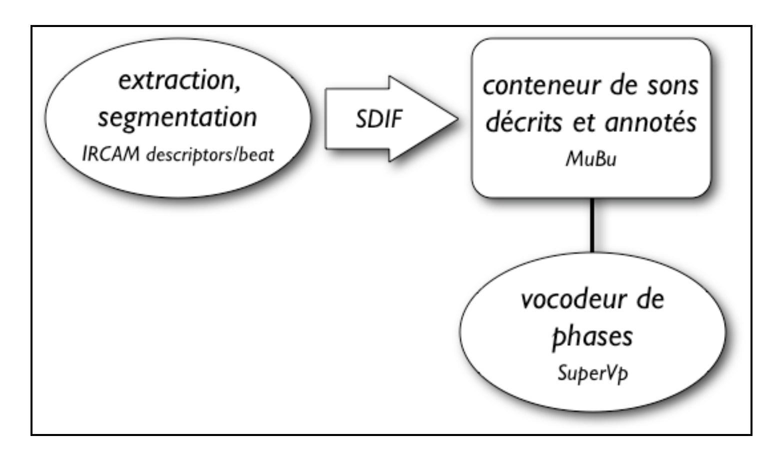


Figure 64 - Ensemble de modules pour morphing sonore et les transformations temporelles interactives.

L'implémentation Max/MSP de l'application s'appuie sur les modules développés dans le cadre du SP3-5, **`sor.sync~`**, **`supervp.cross~`** et **`sor.mubu`**.

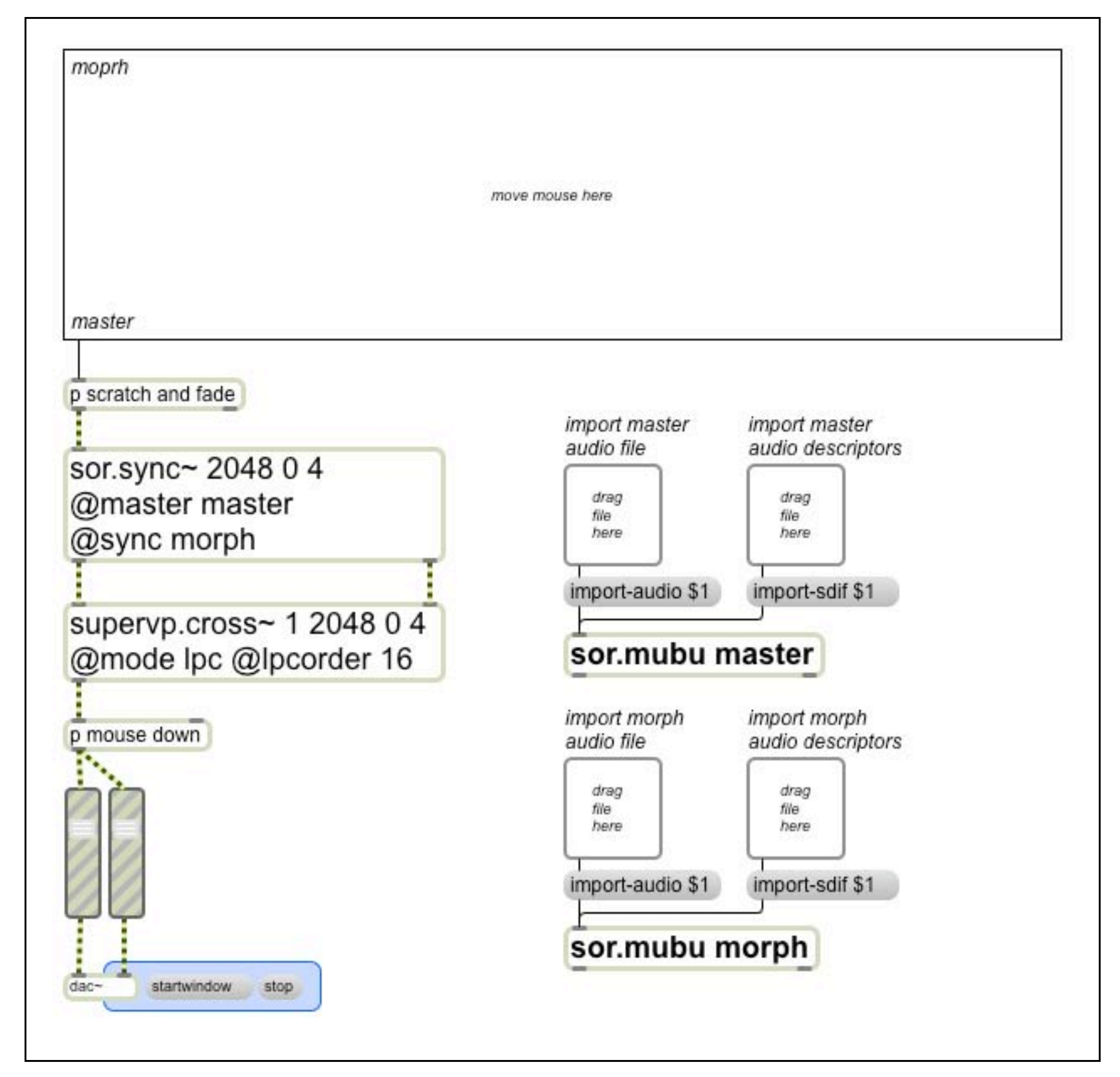


Figure 65 - Application Max/MSP 5 de transformations temporelles interactives et de morphing sonore.

Intégrant le moteur *SuperVP*, le module **`sor.sync~`** permet le balayage (« scrubbing ») synchronisé sur deux fichiers audio (« master » et « sync ») chargé dans deux instances *MuBu* Max/MSP utilisant le module **`sor.mubu`** (SP3-1). En plus des échantillons audio, une segmentation alignée est chargée dans chacun des modules *MuBu*. Cette segmentation définit le même nombre de segments pour chaque fichier audio de manière à créer une correspondance approximative entre les deux segmentations (e.g. s'appuyant sur une notion de notes, de syllabes ou similaire). Le module **`sor.sync~`** est contrôlé par une position relative dans les fichiers audio pour pouvoir reproduire la temporalité de l'un ou de l'autre fichier ou une temporalité hybride.

Dans l'application réalisée, les deux sons synchronisés, sont ensuite traités par le module **`supervp.cross~`** qui permet de transformer et mélanger les timbres pour réaliser l'effet de « morphing » souhaité.

2.3 Echantillonneur

Responsable : Univers Sons

2.3.1 SoundBrowser

Afin de tester et valider les différents scénarios liés à la recherche de sons dans de grandes bases de données, nous avons créé une application dédiée faisant office de prototype d'expérimentation.

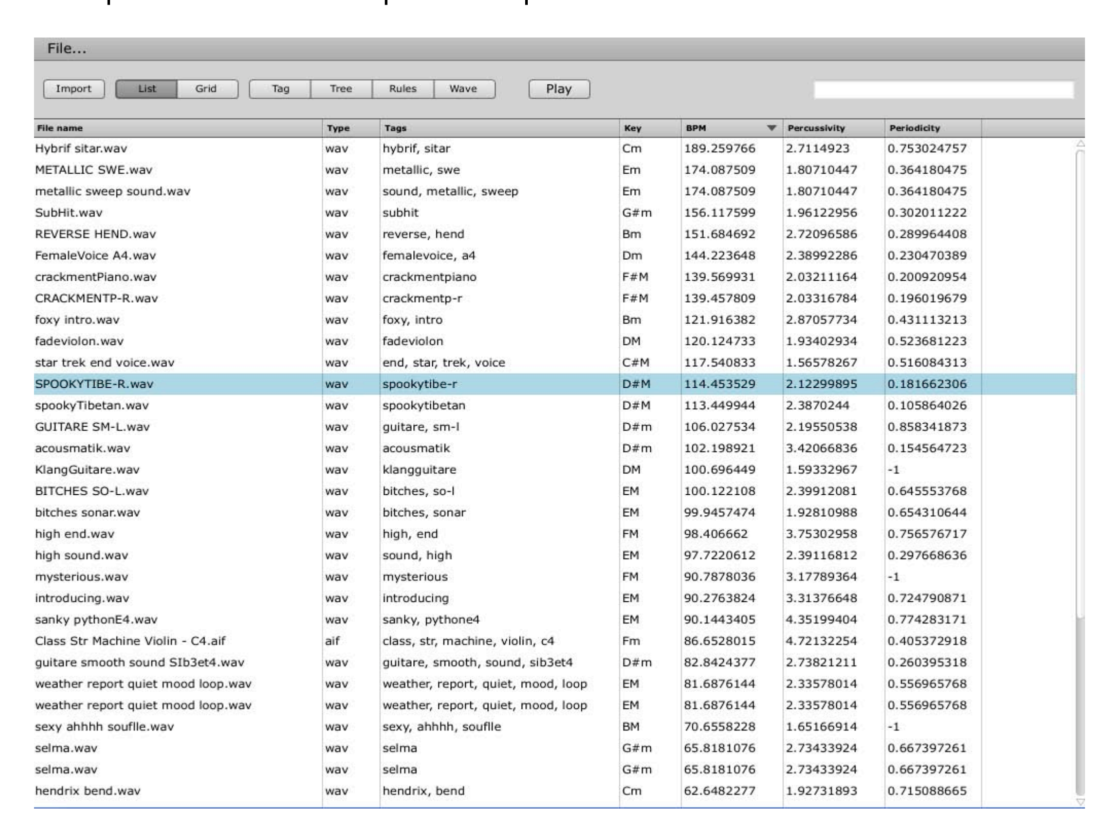
Cette application a servi à la fois de réceptacle pour l'intégration des technologies d'indexation de l'Ircam et de test grandeur nature des différentes possibilités de présentation et d'interaction avec les résultats.

2.3.2 Affichage traditionnel sous forme de liste

L'affiche traditionnel des résultats de recherche se présente sous la forme d'une table où chaque ligne représente un résultat et chaque colonne un attribut du résultat tel que le nom, le nombre de canaux, la tonalité...etc.

Ce type d'affichage permet d'afficher un grand nombre d'informations et permet aux utilisateurs d'avoir un sentiment de maitrise des contenus proposés grâce au coté exhaustif des informations affichées.

De plus les différentes colonnes de la table peuvent être rajoutée ou enlever à volonté afin de personnaliser l'affichage en fonction du type de recherche. Il est ainsi aisé de proposer différentes présélection de vues par défaut pour les utilisateurs novices.



File name	Type	Tags	Key	BPM	Percussivity	Periodicity
Hybrif sitar.wav	wav	hybrif, sitar	Cm	189.259766	2.7114923	0.753024757
METALLIC SWE.wav	wav	metallic, swe	Em	174.087509	1.80710447	0.364180475
metallic sweep sound.wav	wav	sound, metallic, sweep	Em	174.087509	1.80710447	0.364180475
SubHit.wav	wav	subhit	G#m	156.117599	1.96122956	0.302011222
REVERSE HEND.wav	wav	reverse, hend	Bm	151.684692	2.72096586	0.289964408
FemaleVoice A4.wav	wav	femalevoice, a4	Dm	144.223648	2.38992286	0.230470389
crackmentPiano.wav	wav	crackmentpiano	F#M	139.569931	2.03211164	0.200920954
CRACKMENTP-R.wav	wav	crackmentp-r	F#M	139.457809	2.03316784	0.196019679
foxy intro.wav	wav	foxy, intro	Bm	121.916382	2.87057734	0.431113213
fadeviolon.wav	wav	fadeviolon	DM	120.124733	1.93402934	0.523681223
star trek end voice.wav	wav	end, star, trek, voice	C#M	117.540833	1.56578267	0.516084313
SPOOKYTIBE-R.wav	wav	spookytibe-r	D#M	114.453529	2.12299895	0.181662306
spookyTibetan.wav	wav	spookytibetan	D#M	113.449944	2.3870244	0.105864026
GUITARE SM-L.wav	wav	guitare, sm-l	D#m	106.027534	2.19550538	0.858341873
acousmatik.wav	wav	acousmatik	D#m	102.198921	3.42066836	0.154564723
KlangGuitare.wav	wav	klangguitare	DM	100.696449	1.59332967	-1
BITCHES SO-L.wav	wav	bitches, so-l	EM	100.122108	2.39912081	0.645553768
bitches sonar.wav	wav	bitches, sonar	EM	99.9457474	1.92810988	0.654310644
high end.wav	wav	high, end	FM	98.406662	3.75302958	0.756576717
high sound.wav	wav	sound, high	EM	97.7220612	2.39116812	0.297668636
mysterious.wav	wav	mysterious	FM	90.7878036	3.17789364	-1
introducing.wav	wav	introducing	EM	90.2763824	3.31376648	0.724790871
sanky pythonE4.wav	wav	sanky, pythone4	EM	90.1443405	4.35199404	0.774283171
Class Str Machine Violin - C4.aif	aif	class, str, machine, violin, c4	Fm	86.6528015	4.72132254	0.405372918
guitare smooth sound SIb3et4.wav	wav	guitare, smooth, sound, sib3et4	D#m	82.8424377	2.73821211	0.260395318
weather report quiet mood loop.wav	wav	weather, report, quiet, mood, loop	EM	81.6876144	2.33578014	0.556965768
weather report quiet mood loop.wav	wav	weather, report, quiet, mood, loop	EM	81.6876144	2.33578014	0.556965768
sexy ahhhh souflle.wav	wav	sexy, ahhhh, souflle	BM	70.6558228	1.65166914	-1
selma.wav	wav	selma	G#m	65.8181076	2.73433924	0.667397261
selma.wav	wav	selma	G#m	65.8181076	2.73433924	0.667397261
hendrix bend.wav	wav	hendrix, bend	Cm	62.6482277	1.92731893	0.715088665

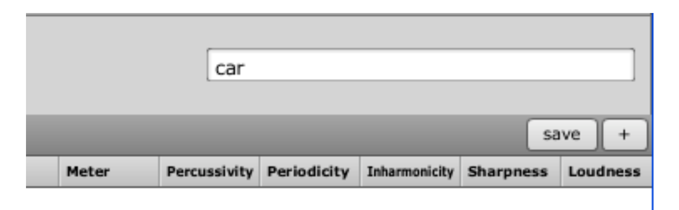
2.3.2.1 Affichage sous forme de grille: mur de sons

Pour compléter l'affichage traditionnel sous forme de liste, dans un souci de simplification et de filtrage de l'information, il nous a paru utile de proposer une vue récapitulative des sons disponibles dans la banque de sons sous forme de mur de sons. Afin de proposer une vue plus agréable et moins technique, plus adaptée à des utilisateurs novices, ou n'étant pas intéressés (dans l'immédiat) par les informations complémentaires proposées.

Ce type d'affichage est très utilisé pour la visualisation de collections d'images et a prouvé son efficacité en terme de communication et d'assimilation de l'information dans un espace réduit.

Afin de fluidifier la navigation dans une grande quantité de résultats, les miniatures de formes d'onde, sont calculées à la demande, avec un système de mise eb cache permettant de d'optimiser les cas de navigation séquentielle linéaire les plus courants.
Un grille 4x3 a été choisie fournissant un compromis entre le nombre de sons visible en seul coup d'œil et le confort d'affichage de chacun des sons permettant d'en discerner les détails.

2.3.2.2 Recherche Textuelle «Full text search»



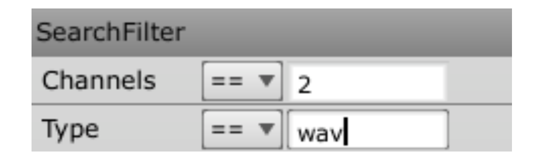
La grande majorité des recherches, que ce soit sur internet ou dans les explorateurs de fichiers fournis avec les systèmes d'exploitation, se limitent à une recherche textuelle pure

avec un ou quelques mots-clés seulement comme en témoigne l'interface minimaliste de recherche Google.

Il s'agit donc d'optimiser au mieux la performance et la pertinence des résultats, tous les autres modes de recherche lui étant subordonnés.

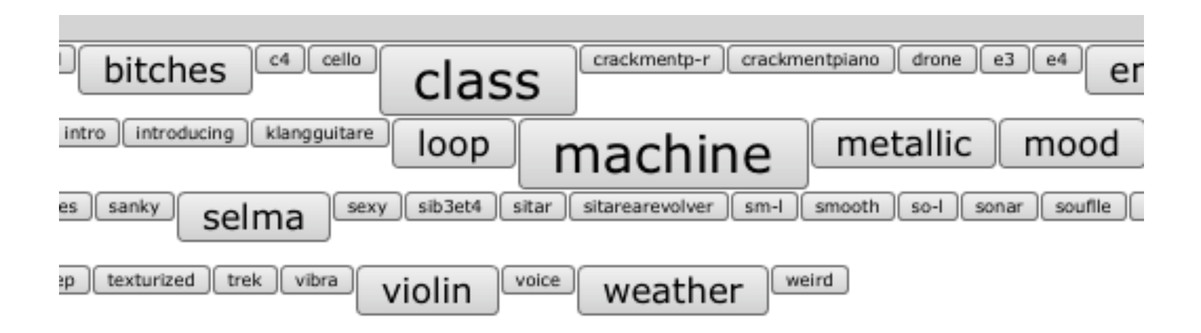
Afin de maximiser les chance de résultat nous avons implémenté une recherche textuelle transversale dite «full text search » avec une table SQL virtuelle permettant de réaliser l'Index inversée mot → sons de manière efficace à la fois en temps de calcul et en espace de stockage.

2.3.2.3 Filtres de recherche classique



Les filtres de recherche traditionnels par accumulation de critères ont été implémentés avec le souci de rester le plus proche possible des outils et habitudes préexistants en terme de recherche en ce qui concerne l'interface afin de familiariser les utilisateurs avec des métaphores et des représentations déjà connues. On trouve ce type de filtres couramment dans le Finder de MaxOSX, dans les clients mails tels qu'Outlook ou Thunderbird.

2.3.2.4 Recherche interactive par tags

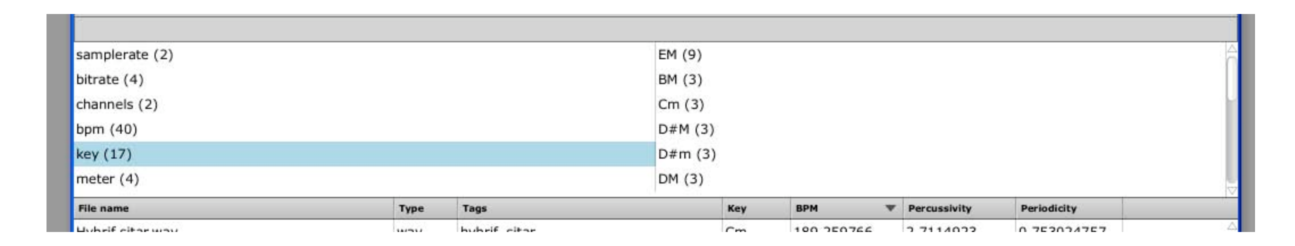


L'affichage sous forme de nuages de mots-clés rendu populaire par son usage sur internet notamment sur les blogs et autres sites communautaires a l'avantage de communiquer une vue à la fois riche et synthétique des résultats dans un espace visuel compact.

De plus, cette visualisation fournit une statistique globale de l'ensemble des sons disponibles dans la base qu'il n'est pas possible d'avoir lorsque l'on affiche des résultats dans un ordre séquentiel comme c'est le cas dans le cas de la liste ou du mur de sons.

L'interface permet d'ajouter successivement les mots-clés aux autres critères de recherche et se rafraîchit au fur et à mesure avec un nouveau nuage à chaque étape permettant d'affiner la recherche par dichotomie. Il est également possible de retirer certains mots-clés si la recherche ne fournit plus suffisamment de résultats, permettant de naviguer de proche en proche entre sous-ensembles de sons.

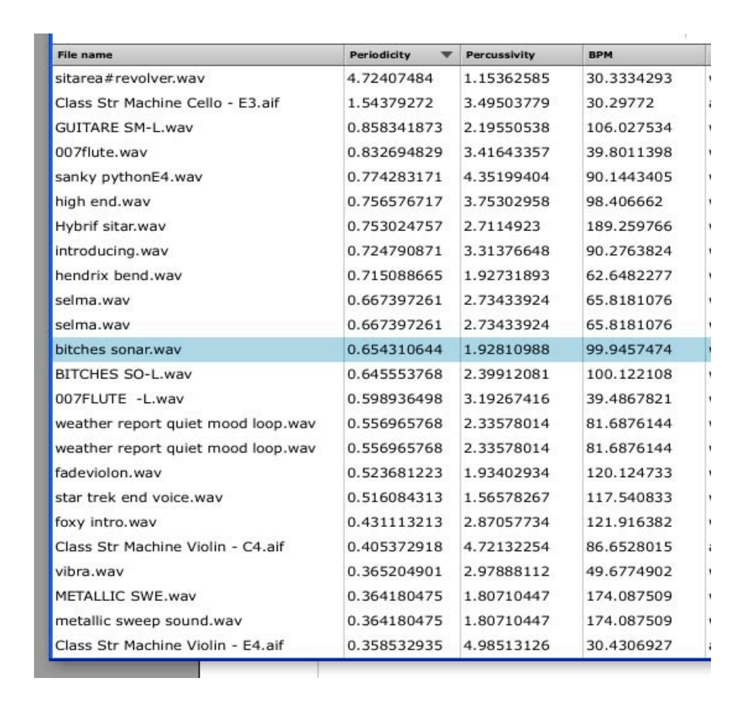
2.3.2.5 Recherche interactive par attributs



De manière similaire à l'affichage par nuage de points, la recherche interactive par attributs n'affiche que les choix réellement disponibles dans la sélection courante ainsi qu'un compteur permettant d'évaluer le nombre de sons partageant le même attribut.

Ici aussi le coté interactif de la visualisation tente d'informer l'utilisateur à la fois d'un information contextuelle, ainsi que d'une statistique globale sur la recherche courante permettant d'avoir une représentation mentale de la répartition des sons selon les différents attributs et catégories.

2.3.2.6 Tri par descripteurs

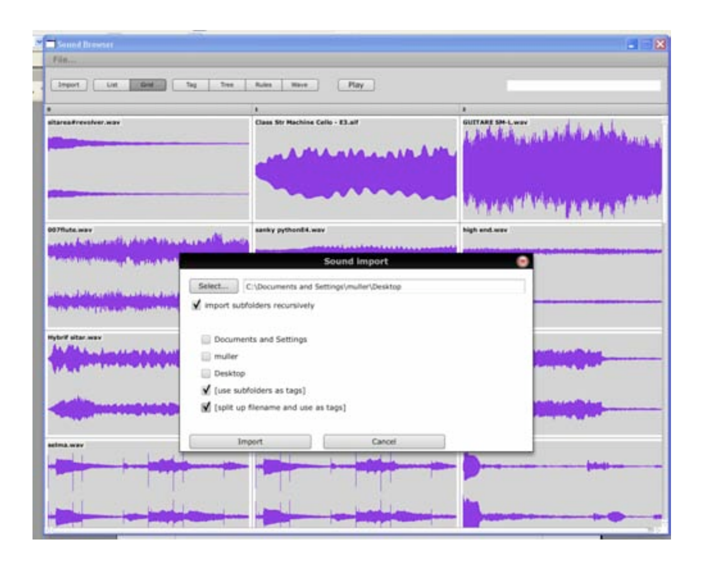


File name	Periodicity	Percussivity	BPM
sitarea#revolver.wav	4.72407484	1.15362585	30.3334293
Class Str Machine Cello - E3.aif	1.54379272	3.49503779	30.29772
GUITARE SM-L.wav	0.858341873	2.19550538	106.027534
007flute.wav	0.832694829	3.41643357	39.8011398
sanky pythonE4.wav	0.774283171	4.35199404	90.1443405
high end.wav	0.756576717	3.75302958	98.406662
Hybrif sitar.wav	0.753024757	2.7114923	189.259766
introducing.wav	0.724790871	3.31376648	90.2763824
hendrix bend.wav	0.715088665	1.92731893	62.6482277
selma.wav	0.667397261	2.73433924	65.8181076
selma.wav	0.667397261	2.73433924	65.8181076
bitches sonar.wav	0.654310644	1.92810988	99.9457474
BITCHES SO-L.wav	0.645553768	2.39912081	100.122108
007FLUTE -L.wav	0.598936498	3.19267416	39.4867821
weather report quiet mood loop.wav	0.556965768	2.33578014	81.6876144
weather report quiet mood loop.wav	0.556965768	2.33578014	81.6876144
fadeviolon.wav	0.523681223	1.93402934	120.124733
star trek end voice.wav	0.516084313	1.56578267	117.540833
foxy intro.wav	0.431113213	2.87057734	121.916382
Class Str Machine Violin - C4.aif	0.405372918	4.72132254	86.6528015
vibra.wav	0.365204901	2.97888112	49.6774902
METALLIC SWE.wav	0.364180475	1.80710447	174.087509
metallic sweep sound.wav	0.364180475	1.80710447	174.087509
Class Str Machine Violin - E4.aif	0.358532935	4.98513126	30.4306927

Grâce aux descripteurs (par exemple brillance, intensité, percussivité...), il est également possible de trier les sons de manière qualitative selon une dimension privilégiée même sans connaître la signification précise des valeurs calculées.

Les tests menés nous ont permis de constater que ce genre de critères permettait d'accélérer la recherche de certaines catégories de sons notamment lorsque les informations textuelles ne sont pas disponibles. Les cymbales et charleston par exemple se regroupent avec de fortes valeurs de brillance alors qu'à l'opposé on trouve assez naturellement des sons de basses beaucoup plus ronds.

2.3.2.7 Import et Analyse



La partie import de nos propres banques de sons ainsi que des banque de sons personnelles constituées par les utilisateurs, nous paraissaient primordiale. De plus, il était important que l'import et l'annotation des sons grâce à la nomenclature des dossiers et noms de fichiers, très rigoureuse dans le cas de nos propres banques de sons, puisse se faire de manière automatisée. Les librairies d'indexation de l'Ircam (IrcamBeat, IrcamKeyMode, IrcamDescriptor) ont également été intégrées afin de permettre l'accès aux informations telles que le tempo, la tonalité, la brillance... etc.

2.3.3 Prototype échantillonneur

2.3.3.1 De Sample Orchestrator à MachFive 3

Parmi les logiciels utilisant le moteur UVI figure le sampler MachFive développé par les équipes d'Univers Sons. Le projet Sample Orchestrator a été envisagé dans un premier temps comme un prototype permettant d'explorer de nouvelles voies dans le domaine du sampling. Au cours du projet, Il nous a semblé naturel de tout d'abord rapprocher puis rassembler ces deux applications pour finalement n'en faire plus qu'une ce qui confère à l'application finale de Sample Orchestrator un fort potentiel de diffusion commerciale.

Sample Orchestrator dépasse les frontières du sampling traditionnel et se positionne comme la première application à offrir une telle diversité se démarquant néttement par rapport aux produits concurrents.

Sample Orchestrator n'est donc plus un simple prototype mais le développement d'une application professionnelle qui sera vendue en magasin avec les contraintes de finition et le niveau d'exigence que cela implique.

2.3.3.2 Description de l'application

L'application MachFive3 est l'aboutissement et l'extension d'un logiciel traditionnellement orienté vers le sampling et la lecture de banques de sons multi-échantillonnés vers une architecture hybride et modulaire à la croisée de(s) la synthèse(s) et du sampling, idéal pour le sound design.

De ce fait, l'architecture du moteur UVI ainsi que l'interface de MachFive ont du être totalement repensées afin de présenter de manière simple et conviviale un nombre illimité de modulations, d'effets, de pouvoir interchanger et combiner différentes formes de synthèse au sein d'un même instrument.

Ces changements ont un impact très fort sur la nouvelle identité visuelle de l'application MachFive3 qui se démarque nettement des 2 précédentes versions MachFive et MachFive2. Ils ont également pour effet positif d'augmenter l'ergonomie de l'application tout en augmentant ses possibilités, sans pourtant dérouter les utilisateurs traditionnels.

L'application de compose de 3 sections principales:

- Un éditeur permettant la création d'instrument.
- Un mixeur permettant de visualiser les niveaux de sorties de chacun des instrument set d'appliquer des effets sur chacuns ainsi que globalement.
- Une partie opérateur permettant de charger et d'utiliser les instruments précédemment créés tout en offrant des contrôles de haut niveau sur ceux-ci.

Nous détaillerons ici la section d'édition qui est le cœur même de l'application

La partie édition comprend:

- un éditeur de mapping qui permet de définir des zones auxquelles on associera des sources sonores (lecteur de son, forme d'onde...) ainsi qu'un ensemble éventuel d'effets et de modulations. Chaque zone est définie par la largeur quelle occupe sur un clavier ainsi que par la plage de velocité qu'elle couvre.
- L'éditeur contextuel de la source sonore actuellement sélectionnée. Chaque source ayant une interface de contrôle dédiée.
- L'éditeur des paramètres liés à cette zone incluant effets, source de modulations et taux de modulation.

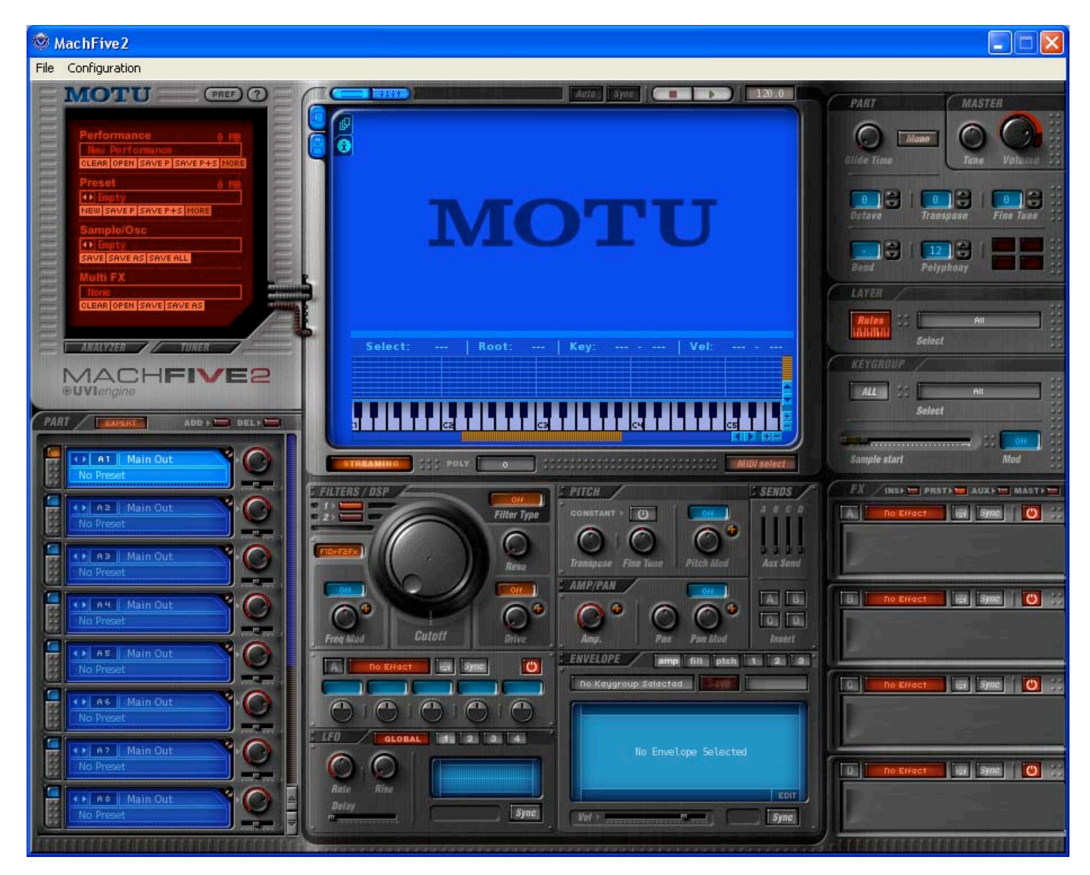


Figure 66 - MachFive 2

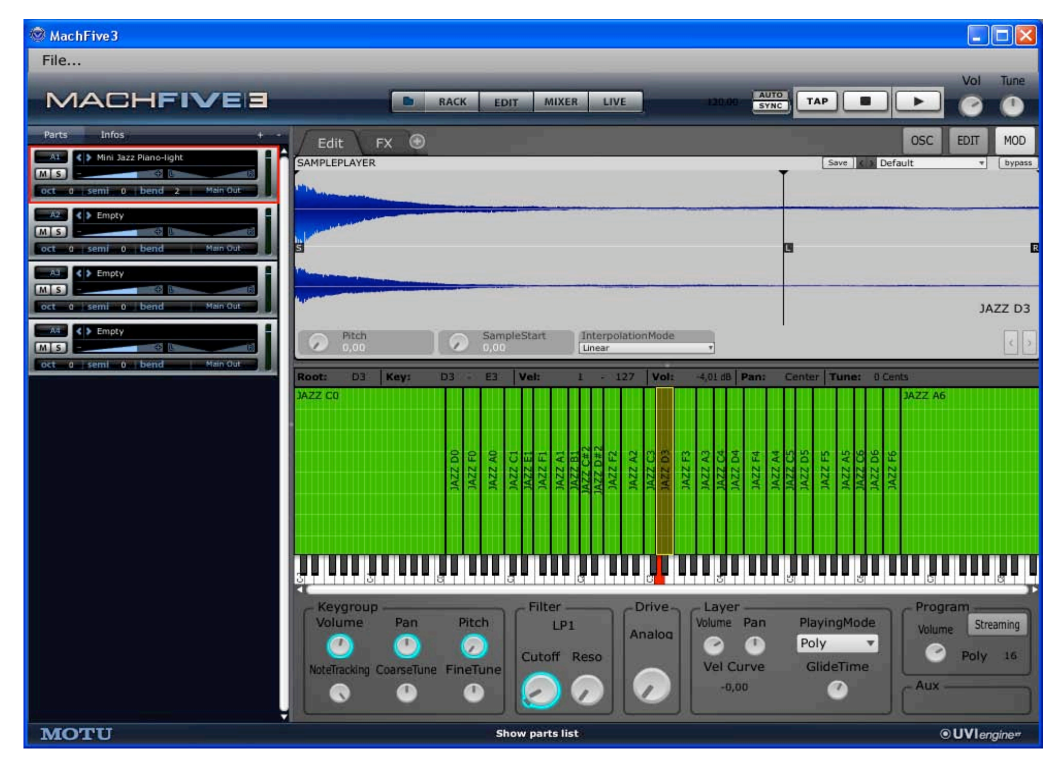
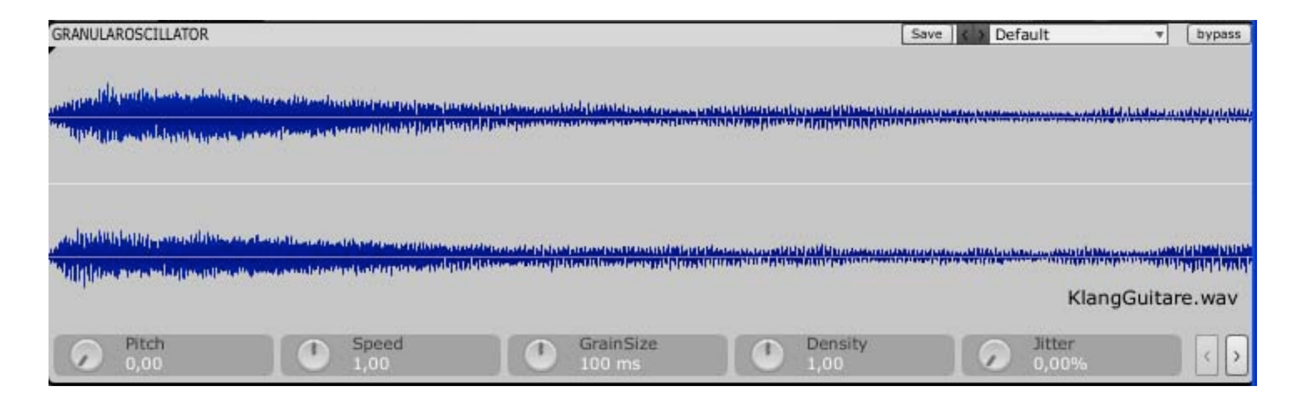


Figure 67 - MachFive 3 – version intermédiaire

2.3.3.3 Détail des oscillateurs novateurs

<u>Synthèse granulaire SOGS</u>



Le module de synthèse granulaire SOGS basé sur les librairies de l'Ircam a été intégré sous forme d'un oscillateur complémentaire aux autres types de synthèses disponibles dans l'application.

La synthèse granulaire de ce type permet de bénéficier d'une vaste palette de textures sonores ainsi que d'une grande flexibilité dans la manipulation de celles-ci, permettant de figer l'évolution temporelle, de naviguer de façon quelconque à l'intérieur des sons et d'effectuer diverses transformations drastiques tout en préservant l'essence du son originel. Les possibilités offertes sont quasi infinies en raison de la base de travail (échantillon sonore) qui peut être interchangeable à souhait.

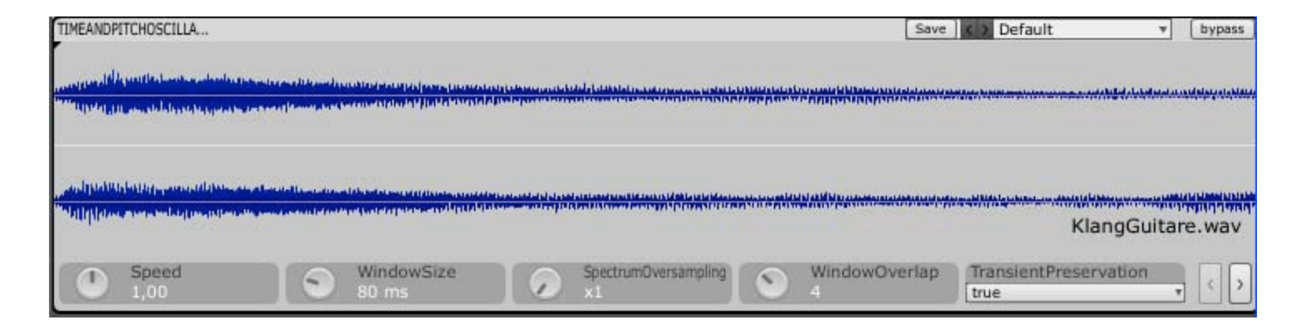
Le contexte applicatif et les catégories d'utilisateurs potentiels étant différents des usages propres à l'Ircam au sein du logiciel Max/MSP, une partie du travail d'intégration a également consisté à adapter les paramètres de contrôle et définir leur plages de variations ou a en combiner certains pour un usage plus intuitif. Nous avons introduit par exemple un contrôle de densité de grains qui pilote lui-même plusieurs autres paramètres de synthèse.

<u>Synthèse granulaire synchrone à la hauteur</u>

Une synthèse granulaire spécifique adaptée aux sons à hauteur monophonique a également été incorporée, ainsi qu'un algorithme spécifique de marquage de forme d'onde avec détection de hauteur.

Ce type de synthèse permet également de rendre totalement malléable la base temporelle des sons enregistrés tout en préservant leurs qualités timbrales lors de la transposition fréquentielle. La grande majorité des banques de sons d'instruments existants étant basée sur des enregistrements de notes seules à hauteur connue, le potentiel d'exploration de ce type d'oscillateur est très grand et ce pour une différence de coût de calcul relativement proche des techniques de sampling traditionnelles.

Transposition par vocodeur de phase



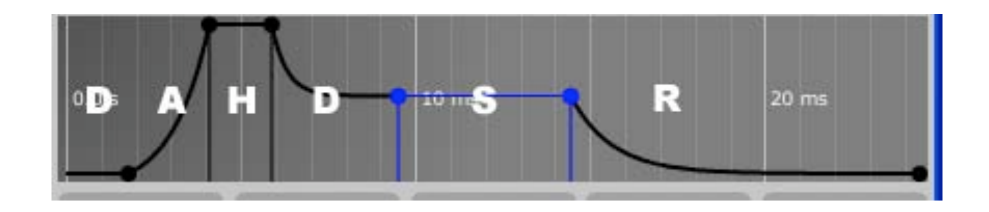
L'algorithme d'analyse-resynthèse SuperVP a également été intégré sous forme d'oscillateur au sein du moteur UVI. C'est l'oscillateur qui offre de loin la meilleur qualité en terme de transformation du son y compris lors de réglages extrêmes comme des dilatations temporelles par un facteur 10 par exemple, grâce à la préservation des transitoires et de l'enveloppe spectrale. Néanmoins cette qualité a un coût qu'il est important de connaître et ne permettra donc pas de jouer un nombre aussi important de notes que les autres types d'oscillateurs plus simples. Mais la qualité exceptionnelle saura contenter les sound-designers les plus exigeants.

Les contraintes spécifiques à Univers-sons, liés à la gestion de la mémoire, du coût de calcul et à sa répartition au cours du temps ainsi que la latence des algorithmes ont été au coeur des échanges entre l'Ircam et Univers-sons durant le projet, permettant d'optimiser les algorithmes en les mettant à rude épreuve.

2.3.3.4 Détail des sources de modulations principales

DAHDSR

Enveloppe classique utilisé notamment pour schématiser l'enveloppe des sons naturels. On peut contrôler le temps d'attaque (A), le temps de maintien (H), le temps de déclin (D), le niveau de soutien (S) ainsi que le temps de relâchement (R) jusqu'à ce que l'enveloppe atteigne sa valeur initiale.



LFO (Low Frequency Oscillator)

Modulation basé sur un générateur basse fréquence

- Forme d'onde: Sinus, carré, Triangle, Rampe,...etc
- Paramètres de fréquence, phase, délai initial et temps de monté.

• Synchronisation au tempo

Ce type de modulation est communément utilisé dans les synthétiseurs pour moduler notamment l'enveloppe d'amplitude (trémolo), ou l'enveloppe de (vibrato).

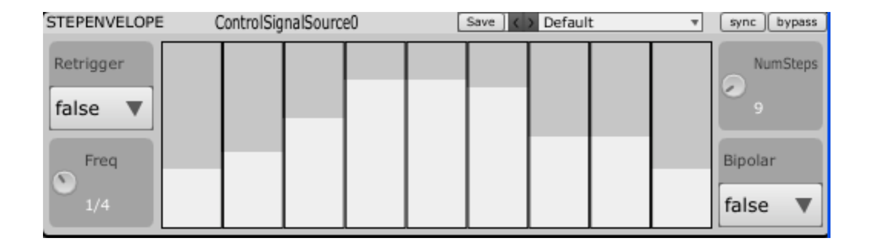


Figure 68 - Formes d'ondes de LFO disponibles

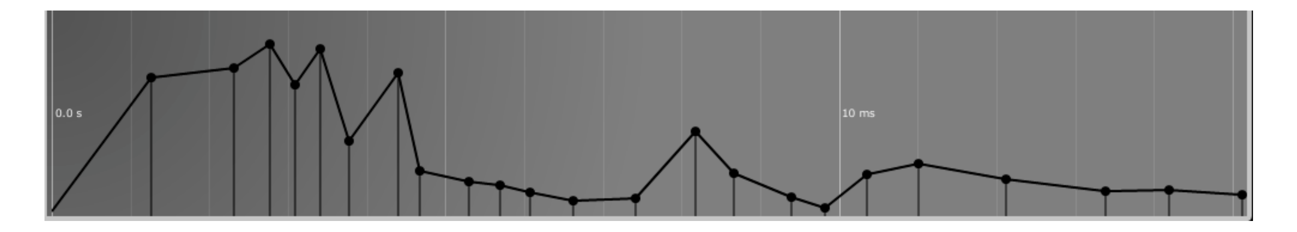
<u>Step Envelope</u>

Modulation basée sur une séquence de niveaux discrets lus séquentiellement communément présentes dans les premiers séquenceurs analogiques des années 70.

Dans notre cas la flexibilité offerte par l'approche logicielle nous permet de définir un nombre de niveaux quelconques ainsi que de pouvoir synchroniser l'horloge interne au tempo courant.



<u>Enveloppe MultiPoint</u>



L'enveloppe Flexible est une source de modulation définie par une suite de points et des courbes concaves ou convexes les reliant entre eux.

Celles ci peuvent être crées par l'utilisateur ou générées à partir d'un échantillon sonore.

Cela permet ainsi d'appliquer cette enveloppe à un tout autre type de son afin de lui conférer une part de la morphologie du son originel.

Afin de simplifier l'interaction ainsi que l'édition ultérieure avec les enveloppes ainsi extraites, un algorithme de simplification de courbe a également été incorporé afin de diminuer le nombre de points nécessaires pour la représentation de la courbe.

La technique de simplification est basée sur l'algorithme de Douglas-Peucker issu à l'origine pour la simplification de tracé au sein des logiciels de Conception graphique assisté par ordinateur.

L'algorithme utilise une approche «top-down» qui consiste à ne conserver que les 2 points extrêmes pour ensuite rajouter un par un les point-clés nécessaire à la minimisation de l'erreur. On est ainsi certains d'avoir uniquement les points nécessaires et optimaux à la description globale de la forme. L'algorithme s'arrête soit lorsque qu'un critère de tolérance soit lorsque que le nombre maximum de points est atteint.

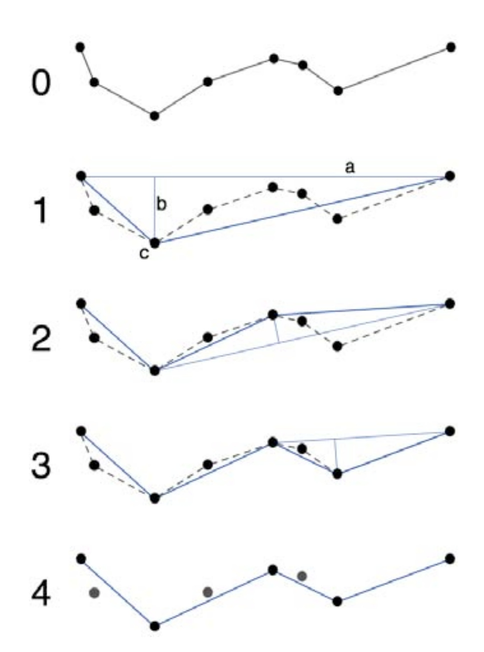


Figure 69 - Algorithme de simplification de tracé de Douglas-Peucker

2.3.3.5 Intégration SoundBrowser dans MachFive3

Grace au prototype SoundBrowser nous avons pu tester et évaluer les possibilités de recherche nouvelles offertes par les techniques d'indexation ainsi que divers scénarios et visualisation des résultats.

Pour l'intégration dans MachFive3, il nous reste à élargir la base de donnée pour pouvoir effectuer non plus seulement des recherches sur des sons isolés, mais également sur d'autres types de contenus de plus haut-niveau lisibles par MachFive, tels que les instruments (constitués de plusieurs centaines de sons), les banques de sons, les séquences musicales... etc

Ce travail dépasse le cadre strict du projet Sample Orchestrator , mais représente une étape indispensable en ce qui concerne l'intégration dans le produit final. Il reste donc à repenser l'interface de recherche pour tenir compte des scénarios de recherche complémentaires tout en piochant dans l'application prototypes toutes les idées dégagées durant le projet.

Vraisemblablement l'interface de recherche devrait de scinder en 3 couches organisées par degré de complexité et niveau d'expertise utilisateur:

- la première (actuellement la seule disponible dans MachFive2) représentant une arborescence de fichiers traditionnelle basée sur notre système de fichiers virtuels pour garder la compatibilité avec nos produits précédents.
- La seconde orientée vers une recherche textuelle simple agrémentée d'un tag-cloud avec un affichage des résultats concis sous forme de liste.
- La troisième plus proche du prototype Soundbrowser, avec un affichage riche sous forme de table ou de miniatures avec les filtres de recherche avancés décrits plut haut.

2.4 Aide à l'orchestration

Le prototype d'orchestration final se présente sous forme de deux composants interopérables à travers le protocole OSC : le serveur d'orchestration Orchidée et son client principal développé dans l'environnement OpenMusic. Ces composants dans leur version finale sont décrits dans les paragraphes suivants.

De plus, cette conception serveur permet la réalisation d'autres clients, comme le prototype de test ci-après réalisé dans l'environnement Max/MSP (version 5).

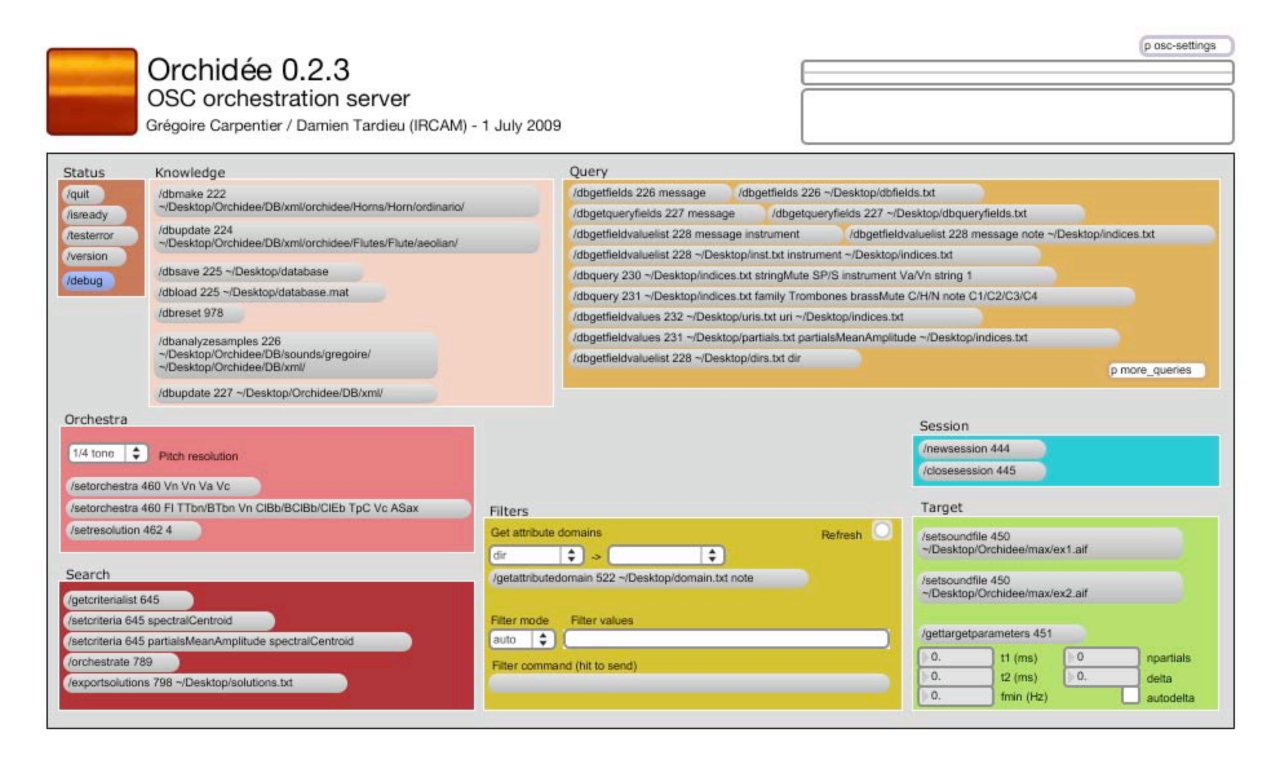


Figure 70 - Interface cliente de test en Max /MSP

2.4.1 Serveur d'orchestration Orchidée

<u>Connaissance instrumentale</u>

La base de données instrumentales utilisée dans le prototype de thèse comptait 11 instruments pour 4763 échantillons. Elle a été étendue à 22 instruments, avec un total de 8 sourdines et 38 modes de jeu. La nouvelle base compte aujourd'hui 17921 échantillons pour une occupation en mémoire d'environ 12 Go. Cette nouvelle base résulte de la réunion de six banques de données disponibles à l'IRCAM :

- -Studio On Line (Sol) v.2.0 ;
- -Vienna Symphonic Library (VSL) ;
- -RWC;
- -Iowa ;
- -Virtual Orchestra ;
- -une banque de sourdines de cuivres fournies par Yan Maresz.

N'ont été retenus, au sein de ces diverses banques de sons, que les échantillons analysés par Damien Tardieu au cours de sa thése. L'ajout de nouveaux échantillons est toutefois possible, à condition bien sûr de fournir les fichiers de métadonnées associés à chacun d'entre eux (cf. infra). La réunion de ces six banques de sons en une seul base homogéne a consisté en les étapes suivantes :

1. vérification systématique du contenu des banques de sons et des métadonnées mises à disposition par Damien Tardieu ;
2. correction des éventuelles inconsistances ;
3. choix d'une nomenclature unique pour l'ensemble des bases ;
4. renommage des échantillons et de métadonnées selon cette nomenclature ;
5. fusion des six banques en un seul système de fichiers ;
6. vérification de la consistance finale.

Les métadonnées associées à chaque échantillon ont été stockées dans un ensemble de fichiers XML. Chaque échantillon de la base est associé à un fichier XML de m^eme chemin et dont le nom est assorti d'un horodatage. Cet horoda-tage sera utilisé comme une information importante pour la mise à jour de la

connaissance instrumentale dans le prochain livrable.

Exemple : l'échantillon /bassoon/ordinario/Bn-ord-A3-mf.wav est accompagné du fichier /bassoon/ordinario/Bn-ord-A3-mffi20090303202113.xml, créé le 3 mars 2009 à 20h21'13".

Notes importantes

1. La banque Sol v.2.0 a été préférée à Sol v.3.0 car les métadonnées d'analyse fournies par Damien Tardieu portent sur la premiére.

2. La base actuelle ne contient aucun échantillon en quart de ton, huitiéme de ton, etc. Afin d'alléger la quantité de données, il a été convenu de ne conserver que les échantillons en demi-tons, à partir desquels le timbre des notes microtoniques pourra être déduit.

3. La justesse des échantillons n'est pas garantie et devra faire l'objet d'une vérification future.

4. Les inconsistances et erreurs de nommage dans les fichiers sons et métadonnées associées ont été éliminées autant que faire se peut. Mais l'erreur zéro n'existe pas. Il conviendra donc, lors l'utilisation future de l'outil, de mettre en place une procédure permettant de corriger, lorsqu'on les découvre, les erreurs subsistantes.

<u>Architecture du serveur</u>

Afin de faciliter l'intégration des outils d'orchestration dans différents environnements (tels qu'OpenMusic ou Max/MSP), une architecture client/serveur a été retenue. Cette organisation permet d'encapsuler à l'intérieur d'un serveur autonome les tâches d'analyse et de calcul, ainsi que l'ensemble de la connaissance instrumentale. En revanche, la conception d'interfaces pour la construction et l'analyse de cibles ainsi que la navigation dans les espaces de solutions sont laissées à la discrétion des environnements clients. Est décrite sur la Figure 71 l'architecture du serveur. Celui-ci est composé de deux objets principaux :

- A gauche, la connaissance instrumentale (objet Knowledge) provient d'une banque d'échantillons sonores associée à une banque de métadonnées sous forme de fichiers XML. Une procédure de mise à jour permet de convertir l'ensemble de ces fichiers en un objet interne unique contenant unedescription symbolique et timbrale des sons. Cette description interne dépend uniquement du type d'information contenue dans les fichiers XML de métadonnées. L'objet Knowledge comprend en outre des méthodes de requête permettant d'en interroger rapidement le contenu,

ainsi que des méthodes permettant de prédire les descripteurs d'une combinaison d'échantillons et le les comparer aux descripteurs de la cible.

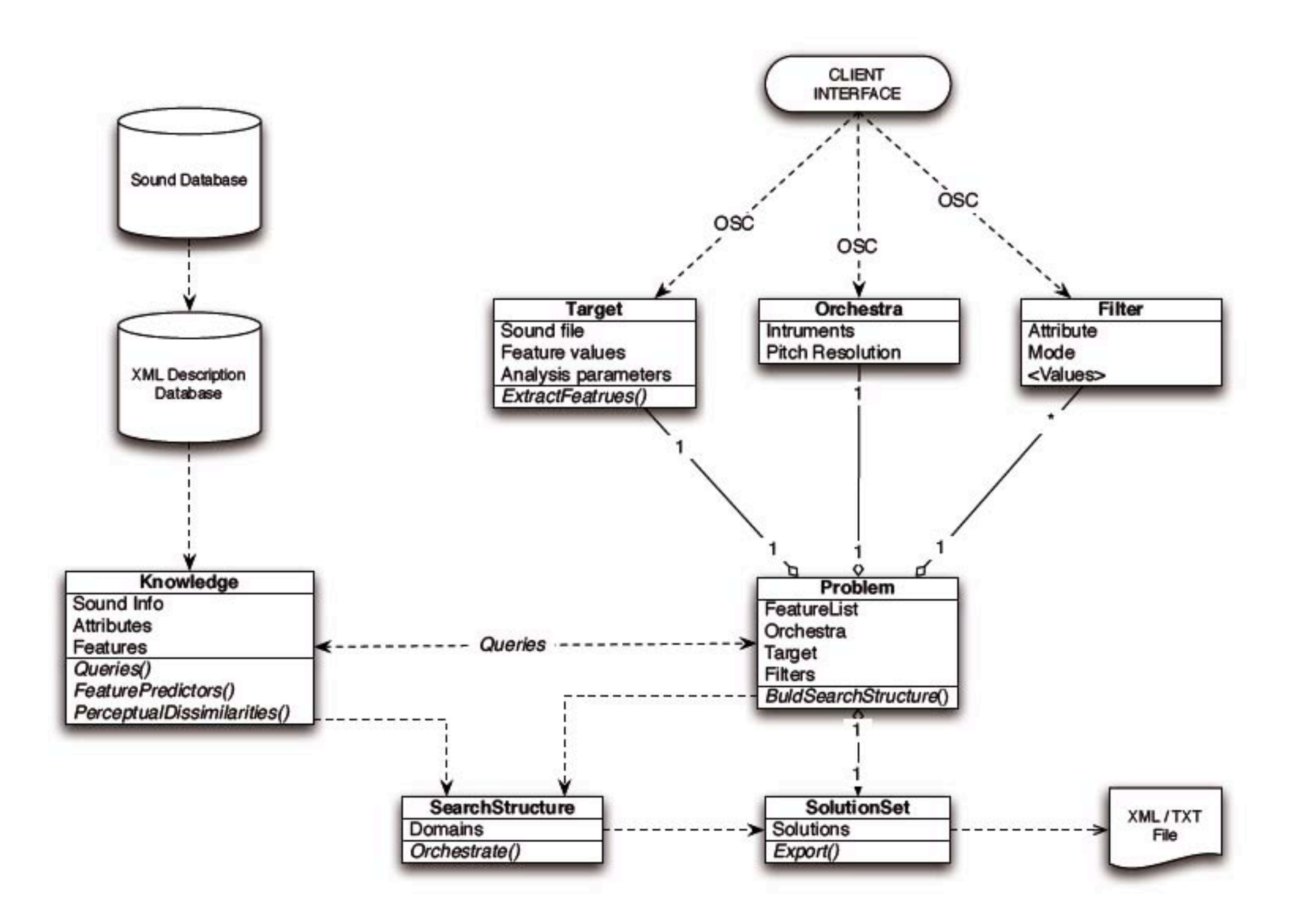


Figure 71 - Architecture du serveur d'orchestration

- A droite, le problème d'orchestration courant (objet Problem) est constitué d'un orchestre, d'une cible et d'un ensemble de filtres permettant de restreindre l'espace de recherche. Ces trois éléments sont construits et modifiés depuis l'interface cliente via un ensemble de messages OSC. L'objet Problem dispose par ailleurs d'une méthode permettant de construire une structure de recherche (objet SearchStructure) comprenant les domaines de chaque variable du problème courant. C'est dans cet objet que l'algorithme d'orchestration sera implémenté. Il fait appel à l'objet Knowledge pour évaluer les propositions d'orchestration. Lorsque l'algorithme se termine, un objet Solution Set est instancié, et peut être exporté sous forme de fichier XML ou texte sur une requête de l'environnement client.

Le serveur d'orchestration communique avec son environnement client via un ensemble de messages OSC. La Figure 72 illustre ce processus d'interaction. Un protocole de communication a été défini afin de permettre une interaction optimale. Il est constitué de message OSC répartis en cinq catégories :

Les messages d'état permettent de renseigner l'environnement client sur l'état du serveur (prêt à faire un calcul, en cours de calcul, accusé de réception d'un ordre) et éventuellement de modifer cet état (réinitialisation, arrêt du serveur).

Les messages de requêtes permettent d'interroger le serveur d'une part sur la connaissance instrumentale dont il dispose, d'autre part sur l'ensemble de solution pour le problème courant.

Les messages de spécification de problème permettent de spécifier un orchestre, un son cible, un ensemble de paramètres d'analyse ainsi qu'un ensemble de filtres permettant de restreindre le domaine de recherche.

Les messages de calcul déclenchent des processus d'analyse ou de recherche combinatoire.

Enfin, les messages d'erreur permettent de signaler une faute de syntaxe ou une impossibilité d'exécution. Ils font par ailleurs remonter vers l'interface client toute erreur interne survenue au sein du serveur, afin d'en faciliter le débogage.

Mise à jour du prototype

Une partie du protocole de communication introduit à la section précédente a été implémentée dans le prototype d'orchestration développé au cours de la thèse. Il existe donc désormais une version serveur de ce prototype, entièrement contrôlable à l'aide de messages OSC. Il est notamment possible, depuis une interface client :

-d'interroger le serveur quant aux instruments disponibles dans sa banque de données ;

-de spécifier la composition de l'orchestre ;

-de spécifier un fichier son (timbre cible) à analyser, ainsi qu'un certain nombre de paramètres d'analyse ;

-d'extraire les descripteurs d'un son cible ;

-de restreindre l'espace de recherche à un ensemble de hauteurs donné ;

-de lancer l'algorithme de recherche d'orchestrations ;

-d'exporter les solutions d'orchestration dans un fichier texte.

La version -serveur -du prototype d'orchestration actuel a été testée à l'aide d'une interface client élémentaire en Max/MSP. La communication avec les interfaces client développées dans OpenMusic par Jean Bresson est en cours de test la rédaction du présent rapport fin juin 09.

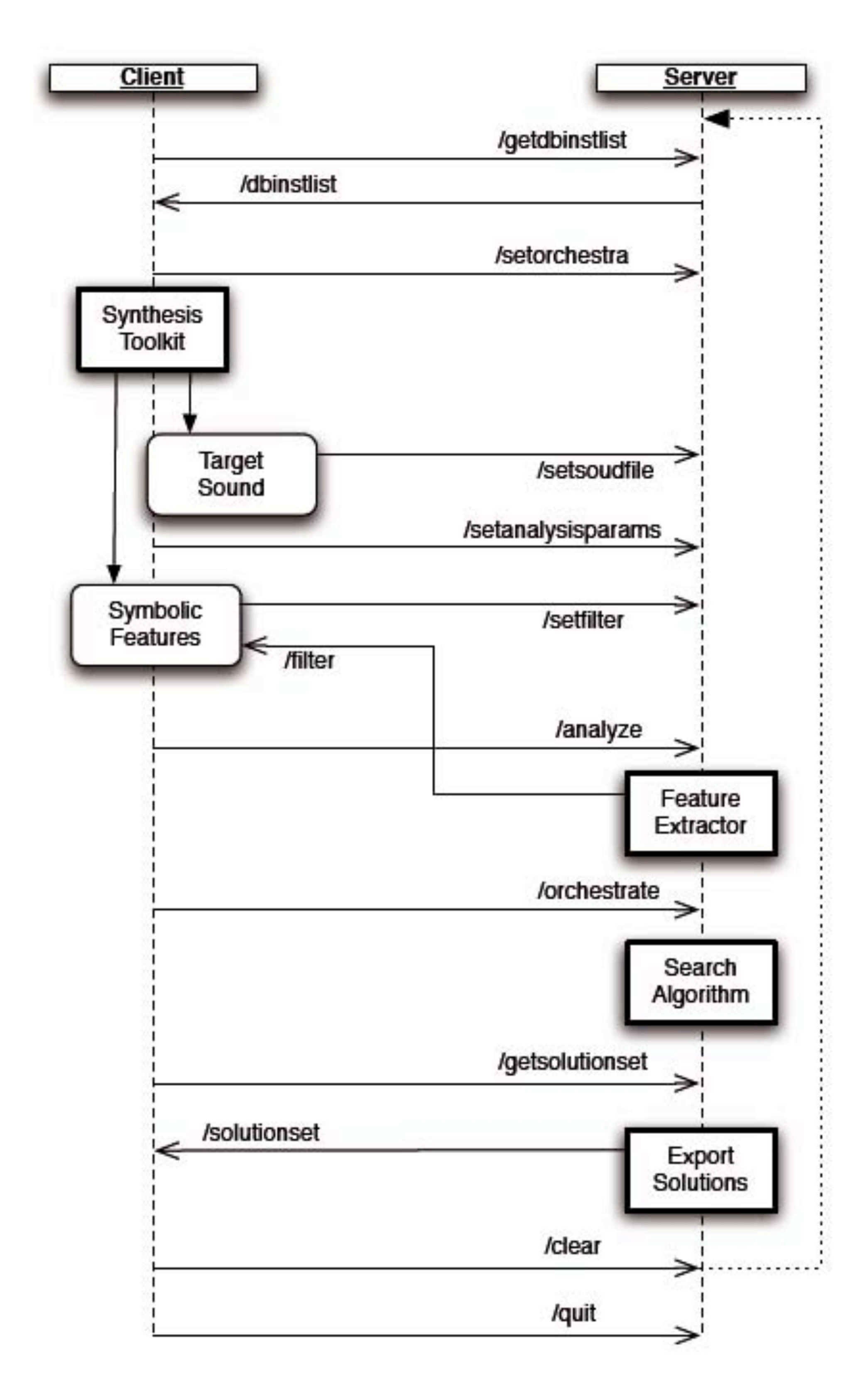


Figure 72 - Scénario d'interaction client/serveur

<u>Transmission</u>

Une séance de formation destinée aux réalisateurs en informatique musicale de la pédagogie a été tenu le 4 février 2009 de 9h30 à 13h en studio B. Etaient présents Mikhail Matl, Eric Daubresse, Grégoire Lorieux, Jean Bresson et Grégoire Carpentier . Suite à cette

séance, les compositeurs du cursus ont pu s'emparer du prototype actuel et en explorer les potentialités. Mikhail Malt a indiqué récemment l'équipe RepMus qu'il était trés satisfait de l'utilisation de ce prototype aveles compositeurs du cursus.

Architecture complétée

Le serveur d'orchestration Orchidée a été implémenté à quelques modifications prés suivant le schéma d'architecture présenté précédemment. Le schéma actuel du système est reproduit sur la Figure 73.

Au niveau supérieur, les interfaces clientes interagissent avec le serveur essentiellement par échange de messages OSC. La communication par fichiers texte est également possible dans les cas où les données à transmettre sont trop volumineuses.

Les messages d'état (Les messages d'état permettent de renseigner l'interface cliente sur la l'état courant du serveur. Leur particularité est de ne pas avoir à accéder aux données manipulées par Orchidée.) sont traités directement par l'interface la plus externe du serveur. Les autres sont redirigés vers un parser qui les interprête selon le type d'objet qu'ils concernent : knowledge ou session.

L'objet knowledge

L'objet knowledge est la représentation interne des connaissances musicales utilisées par Orchidée pour chercher des propositions d'orchestration. On peut y accéder gr^ace à des méthodes de type build ou query. Les premiéres permettent de lire une ensemble de fichiers de description XML et d'en représenter le contenu dans un objet Matlab. Des techniques élémentaires de compression basées sur la détection de redondances en restreignent l'occupation en mémoire. Les secondes permettent un accés au contenu de l'objet knowledge à travers un ensemble de requêtes pour lesquelles une syntaxe a été définie. Un ensemble de tables d'index garantit par ailleurs un accés en O(1) pour les requêtes élémentaires.

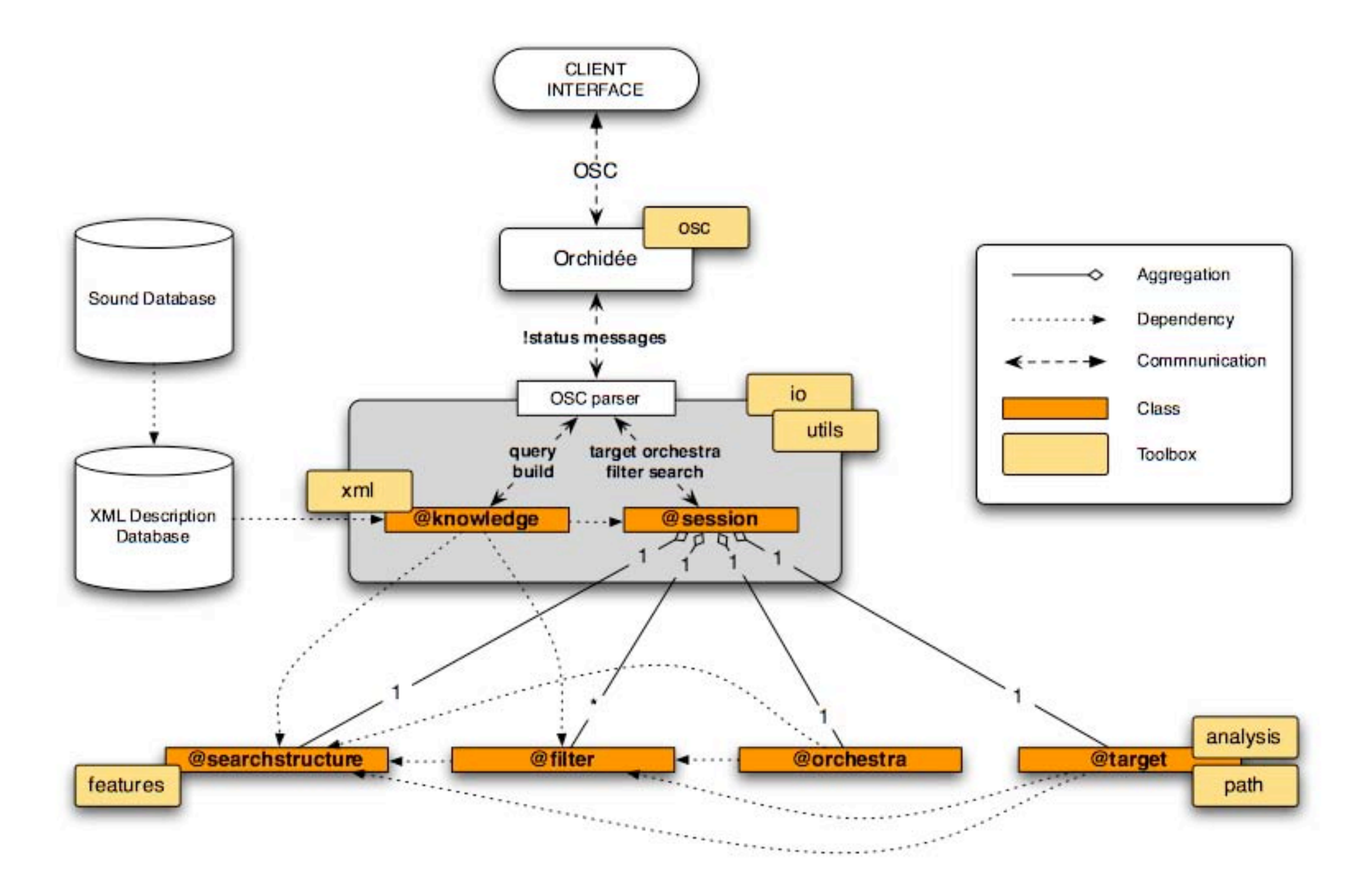


Figure 73 - Architecture générale du serveur d'orchestration Orchidée

L'objet session

L'objet session rassemble l'ensemble des données relatives au problème d'orchestration courant : orchestre, cible, filtres, structure de recherche. Les trois premiers peuvent ^etre édités directement par l'interface cliente. Ils générent ensuite une structure de recherche (objet searchstruture) qui contient les données locales (espaces de recherche et matrices de descripteurs) sous une forme adaptée
à une manipulation algorithmique optimale dans Matlab.

Extensibilité de la connaissance

Au cours de la conception et du développement d'Orchidée, un effort particulier a été fait pour faciliter l'extension de la connaissance instrumentale. Cela peut être entrepris de deux maniéres :

- par ajout d'échantillons instrumentaux ;
- par ajout de descripteurs.

Ajout d'échantillons

Inclure de nouveaux échantillons dans la connaissance instrumentale consiste simplement à rajouter des fichiers XML de description dans l'arborescence courante. Une méthode de

l'objet knowledge permet alors de scanner les nouveaux fichiers de description et d'actualiser la représentation interne de la connaissance. Avant l'import la conformité de chaque fichier XML avec un patron de description metadata template est vérifiée. L'homogénéité de la connaissance est ainsi garantie.

Ajout de descripteurs

L'ajout de nouveaux descripteurs se fait en quatre étapes :

1. ajout des nouveaux champs et des valeurs correspondantes dans les fichiers XML de description ;
2. mise à jour du patron de description metadata template ;
3. mise à jour de l'objet knowledge (de la m^eme maniére que pour l'ajout d'échantillons) ;
4. implémentation dans la toolbox features des méthodes permettant d'agréger et de comparer les descripteurs nouvellement ajoutés, ainsi que de déduire les descripteurs des hauteurs microtoniques à partir du demi-ton

1. le plus proche. Lors d'utilisation d'Orchidée, les critéres d'optimisation sont choisis parmi les descripteurs présents dans l'objet knowledge et pour lesquels les méthodes de d'agrégation, de comparaison et de transposition microtonique sont implémentés dans la toolbox features.

Coopération avec OpenMusic

Un protocole de communication basé sur un ensemble de messages OSC a été défini en collaboration avec Jean Bresson. Depuis le 1er mars 2009 ce protocole s'est enrichi de nouvelles fonctionnalités et définit aujourd'hui une API haut niveau pour le contr^ole d'Orchidée. Une spécification détaillée de ce protocole est disponible dans la documentation finale d'Orchidée, qui est distribuée au 1er juillet 2009.

La version serveur du prototype actuel d'orchestration2 incorpore une version simplifiée de l'API OSC d'Orchidée. Cela a déjà permis à Jean d'initier le développement d'interfaces clientes dans OpenMusic. D'ici la fin du mois, nous allons concentrer nos efforts sur le contrôle des nouvelles fonctionnalités d'Orchidée dans OpenMusic. Un séminaire Recherche & technologie est prévu le mercredi 27 mai 2009 pour présenter l'avancée de nos travaux.

Ce travail en collaboration avec Jean Bresson a donné lieu à un article accepté pour publication dans le Computer Music Journal : « nteracting with Symbolic, Sound and Feature Spaces in Orchidée, a Computer Aided Orchestration Environment. »

Mise à jour du prototype 2

A la demande de plusieurs compositeurs du cursus, la connaissance instrumentale du prototype actuel d'orchestration a été augmentée. On peut désormais réaliser des orchestrations incluant des quarts de tons de violon, alto et violoncelle, et ce pour tous les modes de jeu initialement disponibles pour les demi-tons. Le compositeur Fernando Villanueva a déjà pu tirer parti de cette amélioration pour des orchestrations de fragments vocaux.

2.4.2 Interface cliente : OM-Orchidée

La bibliothèque OM-Orchidée propose un ensemble d'outils, objets et interfaces dans l'environnement OpenMusic constituant un client du serveur *Orchidée*. La plupart des requêtes et instructions définies dans le protocole de communication du serveur sont accessibles sous forme de boites fonctionnelles pouvant être intégrées dans le cadre d'un processus (ou programme visuel) développé dans l'environnement de CAO.

Parmi ces outils, l'objet orchestra offre une interface à la définition d'un orchestre, et permet de communiquer celle-ci au serveur. Il est également possible de spécifier filtres et contraintes, et de mettre à jour et augmenter la base de connaissance instrumentale à partir d'échantillons sonores supplémentaires.

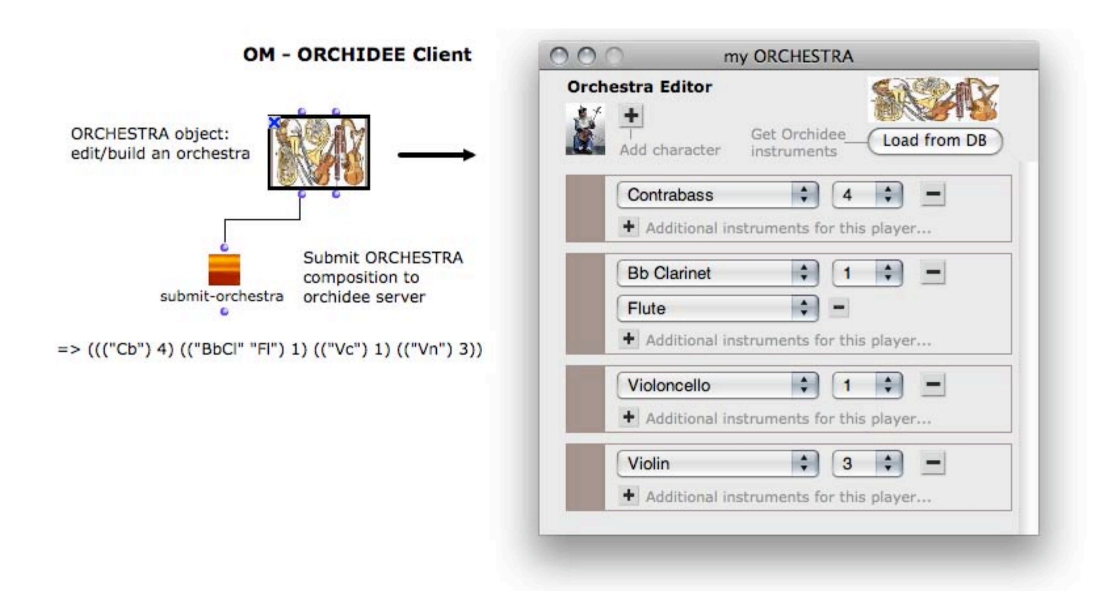


Figure 74 - Objet orchestra dans OM-Orchidée : spécification de l'orchestre

La définition de cibles *abstraites* est une des spécificités du client OM-Orchidée. Cette question en effet, et par opposition à des cibles sonores concrètes ou préenregistrées, constitue un problème complexe ramenant au questionnement fondamental de la conception du son dans l'imaginaire du compositeur. La CAO propose de résoudre ce type de problème en permettant à celui-ci de décrire le son par des processus au travers desquels les descriptions sonores sont construites et transformées, dans une dualité contrôlée entre les aspect compositionnels plus ou moins abstraits et la création des signaux numériques concrets. Ainsi les outils de programmation visuelle permettent de décrire le son suivant différents niveaux d'abstraction, dépendant d'outils et de processus utilisés ou développés par le compositeur. Les structures de description sonore développées dans des systèmes de contrôle de la synthèse comme OMChroma en sont un bon exemple, intégrant des notions statiques et comportementales des représentations sonores, et permettant expansion et calcul dynamiques de ces représentations à partir de spécifications symboliques.

Cependant afin de limiter la part de programmation et pour favoriser une exploration interactive du potentiel sonore, une interface dédiée à la spécification des cibles sonores a été développée dans OM, intégrant représentation graphique spectrale (éditable manuellement) et processus de traitements modulaires issus notament des outuils de composition hérités du système OMChroma et applicables sur cette représentation (génération de composantes spectrales, filtrages, etc.) Cette interface a ainsi pour objectif de permettre au compositeur de déterminer une cible sous forme d'un spectre complexe, dans une démarche compositionnelle expérimentale et interactive, qui masquera les éventuelles complexités de la programmation (l'intérêt étant ici avant tout sur le processus d'orchestration) tout en maintenant les potentialités du système. Cette interface est en réalité un objet (*soundtarget*) qui peut être construit algorithmiquement dans l'environnement de CAO. Les principaux paramètres permettant de construire cet objet sont : 1) un accord, qui permet de spécifier des hauteurs fondamentales et structurantes dans la cible ; 2) des paramètres de filtrage pour un contrôle global de l'enveloppe spectrale ; et 3) des paramètres fonctionnels décrivant le déploiement spectral des différentes hauteurs

fondamentales, c'est à dire des fonctions et paramètres qui s'appliqueront sur les données initiales lors de la construction du spectre.

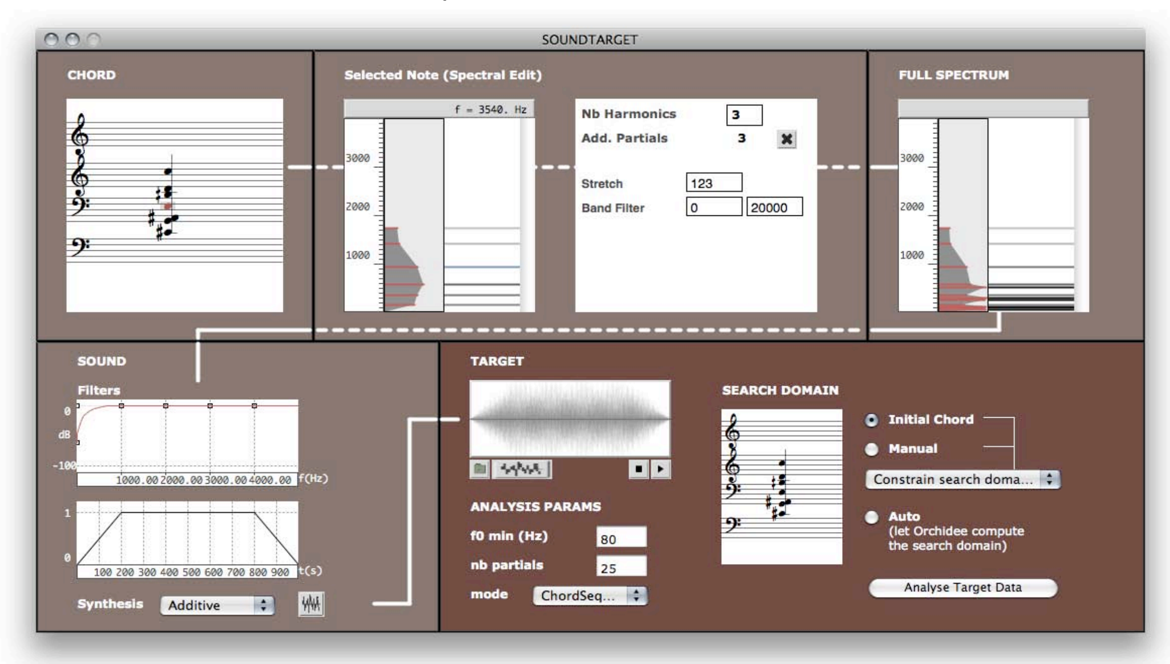


Figure 75 - Interface SoundTarget dans OM-Orchidée : spécification de cibles abstraites

L'interface d'édition de l'objet *soundtarget* propose des représentations symboliques et spectrales de ces différents éléments, et permet à l'utilisateur de les modifier manuellement (ajouter, supprimer, changer la hauteur ou l'intensité des notes et/ou des partiels, les valeurs des paramètres fonctionnels, etc.) et finalement de synthétiser le son correspondant afin d'obtenir un aperçu sonore de la cible spécifiée. Cet objet intègre également des modalités d'instanciation permettant à l'utilisateur de définir les différents modules de filtrages et de traitement/génération de données spectrales, soit à partir d'une bibliothèque de modules prédéfinis, soit à partir de modules programmés (éventuellement sous forme de programme visuel dans OM).

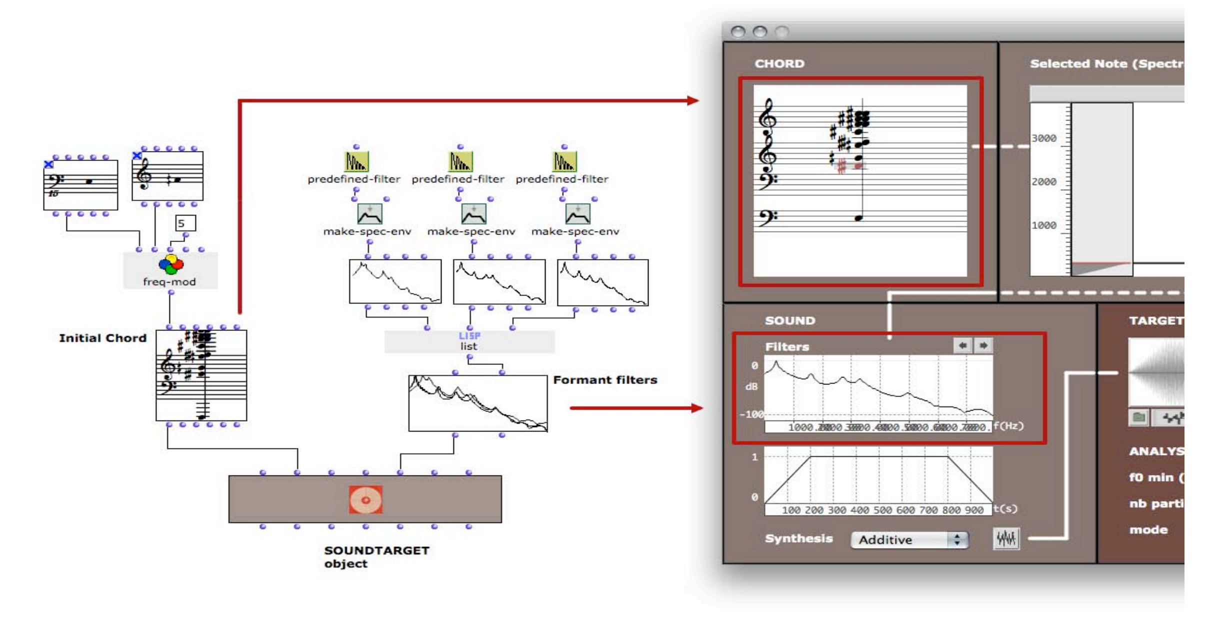


Figure 76 - Génération d'un objet SoundTarget à partir de données symboliques (accord, filtres) dans OpenMusic

Une caractéristique originale de cette interface comparée à un éditeur de spectre classique est en effet due à ces paramètres fonctionnels de l'objet *soundtarget* : la cible est initialement construite à partir de données symboliques et d'opérateurs de traitement de ces données, qui peuvent être issues de fonctions disponibles parmi les fonctions le la bibliothèqiue OM-Orchidée, ou préalablement composées (programmées) par l'utilisateur lui-même dans l'environnement de CAO.

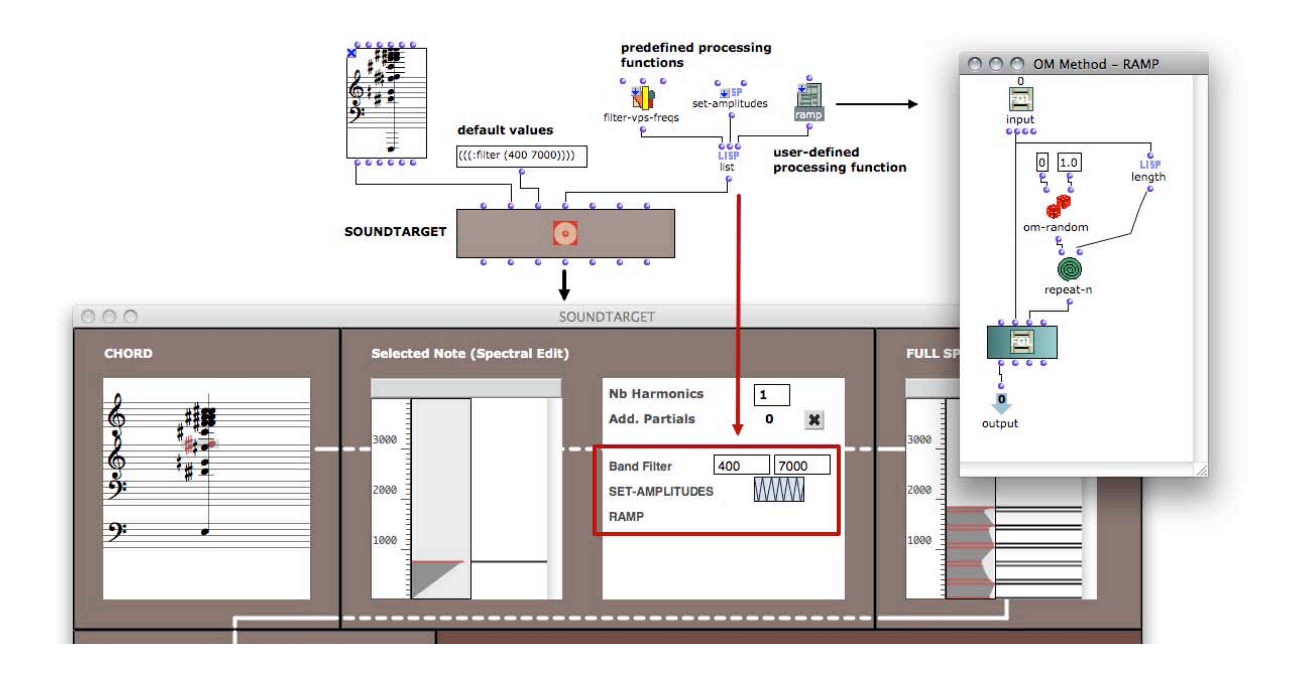


Figure 77 - Traitement algorithmique des données de spécification de la cibles à partir de programmes visuels.

Partant de la cible sonore au sens large (le modèle "abstrait" intégrant potentiellement la notion de cible concrète constituée uniquement d'un son, sans les aspects symboliques et spectraux liés à sa génération), les données utiles (paramètres symboliques, spectraux, le cas échéant, son concret ou synthétisé) sont transmises via le protocole de transfert OSC au moteur d'orchestration qui les utilisera dans sa recherche de solutions.

L'ensemble des solutions est transmis par le serveur sous forme d'un fichier texte, à partir duquel le client OM-Orchidée est également capable de reconstituer des structures de données manipulables dans l'environnement de CAO. Dans l'état actuel du client, diverses manipulations sont proposées pour l'inspection des solutions individuelles, la synthèse sonore par mixage des échantillons de la base de données, ou encore la sélection et la conversion des solutions en structures symboliques ou notation musicale.

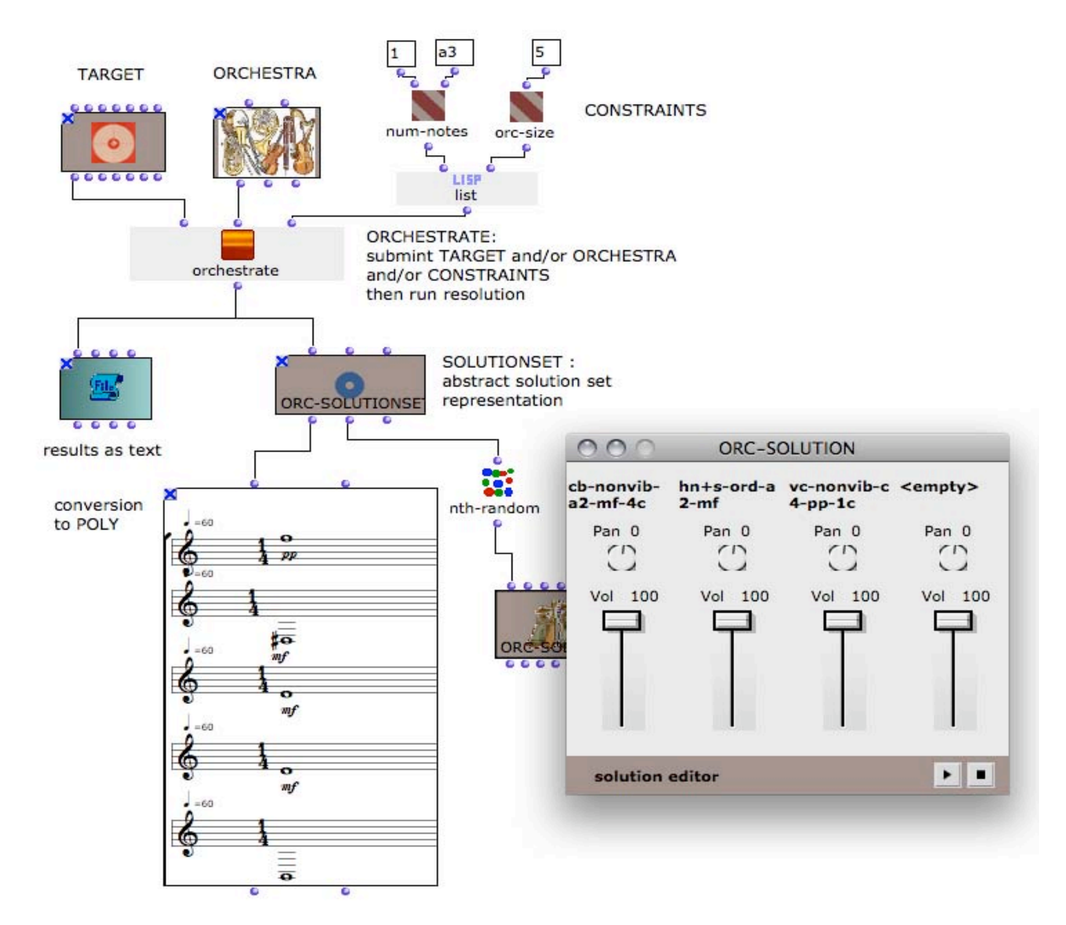


Figure 78 - Utilisation des solutions dans OM-Orchidée: navigation, édition, conversion en notation musicale.

2.4.3 Thèses de doctorat issues du projet

Deux thèses de l'Université Paris 6, école doctorale EDITE, ont été réalisées simultanément dans le cadre du projet et soutenues toutes deux en décembre 2008 : celle de Damien Tardieu et celle de Grégoire Carpentier. Leur contenu est décrit en annexe du présent document.

De plus, un projet de thèse consécutif aux travaux de Master de Philippe Esling (cf § 1.6.11) est soumis dans le cadre d'une demande de bourse à l'école doctorale EDITE pour la rentrée universitaire 2009.

3 Diffusion et valorisation des travaux

3.1 Diffusion scientifique

3.1.1 Publications dans des revues internationales à comité de lecture

[Carpentier09a] Carpentier, G., Bresson, J., « Interacting with Symbolic, Sound and Feature Spaces in Orchidée, a Computer-Aided Orchestration Environment », Computer Music Journal, to appear 2009

[Carpentier09b] Carpentier, G., Tardieu, D., Harvey, J., Assayag, G., Saint James, E., « Constrained Multiobjective Optimization for Computer-Aided Orchestration: Theory and Applications », Journal of New Music Research, to appear 2009

[Carpentier09c] Carpentier, G., Assayag, G., Saint James, E., « Solving the Musical Orchestration Problem using Multiobjective Constrained Optimization with a Genetic Local Search Approach », Journal of Heuristics, to appear 2009

[*article soumis*] Misdariis N., Minard A., Susini P., Lemaitre G., McAdams S., Parizet E., « Environmental sound perception: meta-description and modeling based on independant primary studies », EURASIP Journal on Audio, Speech and Music Processing, special issue on Scalable Audio-Content Analysis (SACA), juillet 2009.

[Roebel08a] Roebel, A. « Frequency-Slope Estimation and Its Application to Parameter Estimation for Non-Stationary Sinusoids », Computer Music Journal, 32:2, pp 68-79, 2008.

[Schwarz07a] Schwarz D., « Corpus-Based Concatenative Synthesis »,*IEEE Signal Processing Magazine*, vol. 24(2), p. 92–104, mars 2007.

[Schwarz08b] Schwarz Diemo, Brunet Etienne, « theconcatenator CD Contributors' Notes. Leonardo Music Journal. Novembre 2008, n° 18

[Zivanovic08a] Zivanovic, M., Röbel, A., Rodet X., « Adaptive Threshold Determination for Spectral Peak Classification ». Computer Music Journal. 32:2, pp. 57-67, 2008.

3.1.2 Chapitres de livres

[Cahen08] Cahen Roland, Jacquemin Christian, Schwarz Diemo, « Navigation et composition dans un univers graphique 3D de grains sonores ». Sophie Chauvin. Information & Visualisation: Enjeux, recherches et applications. Toulouse : Cépaduès Editions. 2008, p. 243-268

[Carpentier07a] Carpentier, G., Tardieu, D., Assayag, G., Rodet, X., Saint-James, E., « An Evolutionary Approach to Computer-Aided Orchestration », EvoMUSART, vol. LNCS 4448, Valence, 2007, pp. 488-497

3.1.3 Actes de colloques avec comité de sélection

[Burred08] J.J. Burred, C.E. Cella, G. Peeters, A. Röbel and D. Schwarz, « Using the SDIF Sound Description Interchange Format for Audio Features», Int. Conf. On Music Information Retrieval (ISMIR), Philadelphia, USA, Septembre 2008.

[Burred09] J.J. Burred, A. Röbel, T. Sikora, « Polyphonic Musical Instrument Recognition Based on a Dynamic Model of the Spectral Envelope », IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Taipei, Taiwan, 2009

[Carpentier06a] Carpentier, G., Tardieu, D., Assayag, G., Rodet, X., Saint-James, E., « IMITATIVE AND GENERATIVE ORCHESTRATIONS USING PRE-ANALYSED SOUNDS DATABASES », SMC'06, Marseille, 2006, pp. 115-122

[Einbond09] Einbond Aaron, Schwarz Diemo, Bresson Jean, « Corpus-Based Transcription as an Approach to the Compositional Control of Timbre ». International Computer Music Conference (ICMC). Montreal, QC : 2009

[Jacquemin07] Jacquemin Christian, Ajaj Rami, Cahen Roland, Ollivier Yoann, Schwarz Diemo, Plumage: Design d'une interface 3D pour le parcours d'échantillons sonores granularisés. Conférence Francophone sur l'Interaction Homme-Machine IHM. Paris : Novembre 2007

[Minard08a] Minard A., Susini P., Misdariis N., Lemaitre G., McAdams S., Parizet E., « Environmental sound description : comparison and generalization of 4 timbre studies », Computer-Human Interface conference / Sonic Interaction Design workshop, Florence, Italie, 2008.

[Minard08b] Minard A., Susini P., Misdariis N., Lemaitre G., McAdams S., Parizet E., « Two-level description of environmental sounds : comparison and generalization of 4 timbre studies », Acoustics'08, Paris, France, 2008.

[Peeters08] G. Peeters and E. Deruty, « Automatic morphological description of sounds », Proc. Acoustics 08, Paris, France, Juin 2008.

[Schnell09] Schnell N., Röbel A., Schwarz D., Peeters G., Borghesi R., « MuBu & Friends - Assembling Tools for Content-Based Real-Time Interactive Audio Processing in Max/MSP », *Proceedings of the International Computer Music Conference (ICMC)*, Montreal, Canada, août 2009.

[Schwarz07b] Schwarz D., Britton S., Cahen R., Goepfer T., « Musical Applications of Real-Time Corpus-Based Concatenative Synthesis », *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, Denmark, août 2007.

[Schwarz08a] Schwarz D., Cahen R., Britton S., « Principles and Applications of Interactive Corpus-Based Concatenative Synthesis », Journées d'Informatique Musicale (JIM), GMEA, Albi, France, mars 2008.

[Schwarz08c] Schwarz D., Cadars S., Schnell N., « What Next? Continuation in Real-Time Corpus-Based Concatenative Synthesis », *Proceedings of the International Computer Music Conference (ICMC)*, Belfast, UK, août 2008.

[Schwarz09a] Schwarz Diemo, Schnell Norbert, Gulluni Sebastien, « Scalability in Content-Based Navigation of Sound Databases ». International Computer Music Conference (ICMC). Montreal : Août 2009

[Schwarz09b] Schwarz Diemo, Schnell Norbert, « Sound Search by Content-Based Navigation in Large Databases ». Sound and Music Computing (SMC). Porto : Juillet 2009

[Tardieu07a] Tardieu, D., Carpentier, G., Rodet, X., « Computer-Aided Orchestration Based on Probabilistic Instruments Models And Genetic Exploration », ICMC'07, Copenhagen/Denmark, 2007

[Tardieu07b] Tardieu, D., Rodet, X., « An Instrument Timbre Model For Computer Aided Orchestration », WASPAA, New Paltz, NY, 2007

[Tardieu07c] D. Tardieu, G. Peeters, X. Rodet. An Instrument Timbre Model For Computer Aided Orchestration. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA, New Paltz, New York, 2007.

[Tardieu08b] Tardieu, D., Carpentier, G., Peeters, G., « Instrument sound description and modelisation in the context of computer aided orchestration », Acoustics 08, 2008

3.1.4 Séminaires et workshops scientifiques

3.1.4.1 Orchestration

G. Carpentier, CDCSolver: Un algorithme de resolution de contraintes dans un systeme d'orchestration assistee par ordinateur, Seminaire du groupe de travail Musique, Contraintes et Interaction (AFIM), Vannes, France, janvier 2008

Présentation au colloque « La complexité dans les arts et la science », 12 juin 2009, Ircam Centre Pompidou

Présentations Séminaire Recherche et Création Ircam

Présentation Séminaire Arts Institut des Sciences Complexes Paris-Ile de France, 2009

3.1.4.2 Synthèse par corpus

Workshop MnM et CataRT aux Journées d'Informatique Musicale (JIM) au GMEA à Albi, 27 mars 2008.
Séminaire R&D sur la synthèse par corpus temps-reel avec CataRT, 30 mars 2008.

3.1.5 Thèses, Masters et autres publications

[Cadars07] Cadars S., « Modélisation temporelle et synthèse concatenative de boucles rythmiques», *Rapport de Stage Master ATIAM*, septembre 2007.

[Carpentier08a] Carpentier, G., « Approche computationnelle de l'orchestration musicale : optimisation multicritère sous contraintes de combinaisons instrumentales dans de grandes banques de sons », Thèse de doctorat Atiam, UPMC, Ircam 2008

Esling, P. « Orchestration musicale dynamique » mémoire de Master II Atiam, Ircam, juin 2009

[Tardieu08c] Tardieu, D., « Modèles d'instrumentspour l'aide à l'orchestration », Thèse de doctorat Atiam / Paris 6, IRCAM-Paris 6, 2008

[Schwarz07c] Schwarz D., « Corpus-Based Synthesis », page web http://imtr.ircam.fr/index.php?Corpus_Based_Synthesis, 2007.

3.2 Diffusion professionnelle

3.2.1 Présentations professionnelles

3.2.1.1 Synthèse par corpus

Les modules et l'application CataRT a été présenté de manière théorique et pratique dans un atelier avec travaux pratique d'une semaine sur FTM&Co. au Bergen Center for Electronic Art (BEK), Bergen, Norvège, 9.–13.3.2009.

FTM&Co et CataRT à été enseigné au stages professionnels week-end de l'Ircam, 24.–25.5.2008 et 6–7.6.2009.

Présentations aux Cursus 1 et 2 des travaux de l'équipe IMTR : 9 janvier 2008 Cursus 2
Présentation aux réalisateurs multimédia du Fresnoy, 16 mai 2008.
Présentations prospective CataRT (synthèse concaténative) stage "Ircam Overview" pour des compositeurs étrangers qui souhaitent avoir une initiation aux outils Ircam sur un mois. (1.7.2009)

3.2.1.2 Autres objets temps réel

Séminaire Prospective sur l'Interaction Musicale, IRCAM, 29/04/2009.

Ateliers et groupe de travail sur la représentation de descripteurs sonores dans Max/MSP, printemps/été 2009.

3.2.2 Forum IRCAM

3.2.2.1 SuperVP

Les améliorations du vocodeur de phase SuperVP ont été partiellement intégrées dans les logiciels du Forum de l'IRCAM. La qualification de l'estimation de la fréquence fondamentale est utilisable dès la version 2.9 de AudioSculpt. Les objets Max/MSP, SuperVP-Max modules, et l'application SuperVP-TraX bénéficient de la transposition en domaine fréquentiel et les autres réductions du temps de calcul apportées dans le cadre du projet Sample Orchestrator ce qui améliore notamment l'utilisation dans le contexte du traitement temps réel. Les nouvelles versions des objets et du logiciel seront distribuées au Forum automne 2009.

Distribution de modules SuperVP pour Max/MSP

- Version 2.11, 4/9/2008
- Version 2.12

Distribution BETA (IRCAM interne), 09/11/2008

- Distribution 2.12.1, 23/11/2008
- Distribution 2.12.3, 22/12/2008

3.2.2.2 Synthèse par corpus

Workshop aux ateliers du Forum: Séances de travaux pratiques avec Catart (13.11.2008).

Diffusion des modules et du prototype CataRT dans le FORUM.

3.2.3 Contribution à la production d'œuvres artistiques

3.2.3.1 SuperVP

Depuis le lancement de la série d'objets Max/MSP *SuperVP Max Modules,* ces outils pour la transformation temps réel ont rencontré un grand succès de la part des réalisateurs en informatique musicale de l »IRCAM. La réduction du temps de calcul et notamment l'indépendance du temps de calcul des paramètres de transformations facilite l'utilisation des objets. Nous ne mentionnerons à titre d'exemples que deux créations d'œuvres en concert qui ont été réalisées lors du festival Agora 2009 :

- Dans la pièce *Prism spectra*, du compositeur Dai Fujikura les objets Max/MSP ont été utilisés pour la création d'un quatuor virtuel a partir d'un seul instrument joué sur scène. Les transpositions de l'alto joué sur scène ont été réalisées avec SuperVP Max.
- Un autre exemple innovateur d'utilisation des objets SuperVP est le projet Antescofo. Ce logiciel de suivi et d'anticipation de l'interprétation d'une œuvre par un soliste utilise les objets SuperVP-Max pour adapter les sons de l'accompagnement au soliste. Un exemple de l'utilisation de Antescofo est le concert *hist* du compositeur Marco Stroppa.

3.2.3.2 Synthèse par corpus

L'installation « Grainstick » par le compositeur Pierre Jodlowski développée dans le cadre du projet européen SAME et créée à l'Ircam en Juin 2009 utilisait CataRT comme moteur granulaire sonore.

L'installation « Xe-Rocks » par l'artiste Cécile Babiole utilise les capacités d'analyse sonore de CataRT et la (re)composition granulaire par navigation dans l'espace des descripteurs.

L'interaction d'un système de synthèse par corpus avec un musicien demande une recherche spécifique tant au niveau du mapping entre le son et des paramètres de contrôle du système qu'au niveau de la constitution du corpus à partir du son live du musicien. Or, ce type d'interaction est très demandée par des compositeurs et des musiciens performeurs de la musique improvisée Dans ce but, des méthodes et stratégies d'interaction avec un musicien ont été étudiées, testées, et appliquées dans plusieurs performances électroacoustiques improvisées. Ce travail a été valorisé dans le cadre d'une performance acceptée au programme musical de la International Computer Music Conference (ICMC), Belfast, en août 2008, puis présentée au concert *Sublime Sonus - Part 2 : La Musique Electroacoustique Live* au centre multimédia *Le Cube* à Issy-les-Moulineaux le 6 décembre 2008 (avec E. Brunet à la clarinette basse et D. Schwarz contrôlant CataRT en temps-réel). En plus, le concept musical et technologique de la constitution temps-réel d'un corpus pour la synthèse concaténative à été sélectionné pour figurer dans le *Leonardo Music Journal* et un morceau d'une performance live antérieure sur le CD l'accompagnant [Schwarz08b]. Diemo Schwarz poursuit cette approche artistique dans des collaborations ad-hoc avec des musiciens tel que Victoria Johnson (violon) dans un concert à la galerie Landmark, Bergen, Norvège le 13.3.2009, et Luka Juhart (accordéon) dans un concert au festival « Earzoom » à la Galérie SKUC, Ljubljana, Slovénie, 29.5.2009.

CataRT est utilisée dans des pièces de musique électro-acoustique suivantes :

- Aaron Einbond, *What the Blind See* for small ensemble and electronics, Cursus 2 concert at Agora 2009, Ircam --- real time control of CataRT from audio analysis and corpus-based transcription
- Roque Rivas *Mutations of Matter*, 2008, pour cinq voix, électronique et vidéo a été réalisé dans le cadre du Cursus 2 de l'Ircam et créé par les Jeunes Solistes (dir. Rachid Saphir) à Agora 2008, en collaboration avec le vidéaste Carlos Franklin, alors étudiant en deuxième année au Studio National-Le Fresnoy. Il figure parmi les lauréats des CONCOURS INTERNATIONAUX DE BOURGES 2009, dans la catégorie "Arts Electroniques-Performance".
- Aaron Einbond, *Beside Oneself* for viola and electronics, Cursus 1 2007–2008, Ircam --- real time control of CataRT from audio analysis of the viola

- Paola Calderone, Cursus 1 2007-2008, Ircam
- Roque Rivas, *Conical Intersect* for fagott and electronics, amongst which CataRT. 6.10.2007; reprise 13.06.2008, Ircam
- David Coll, cursus 1 piece for soprano, petites percussions and electronics, 06.10.2007 Ircam
- Dai Fujikura piece *swarming essence* for Agora 2007 (Ircam), Paris, 9.6.2007
- Stephano Gervasoni piece *Whisper Not* for Monaco, and Agora 2007 (Ircam), Paris, 14.6.2007

3.2.3.3 Autres modules temps réel

« Poetry for // dark -/ dolls », pièce pour soprano, ensemble et électronique de Marco Antonio Suárez-Cifuentes, création IRCAM 10/01/2009 (moteur de « musaicing », synthèse par descripteurs pilotée par l'analyse temps-réel de l'ensemble et du soprano.

3.2.3.4 Aide à l'orchestration

- Speakings pour Orchestre de Jonathan Harvey, première aout 2008 à Londres au Royal Albert Hall, BBC Orchestra
- Fernando Villanueva Carretero,Bukowski Madrigals, cursus 2 création festival Agora 2009
- Marco Suarez Cifuentes Poetry for //dark-/ dolls CM * Cursus 2, création festival Agora 2009 Ircam
- Kenji Sakai Astral/Chromoprojection**, Création Cursus2, Création festival Agora 2009 Ircam

3.3 Diffusion publique, valorisation

3.3.1 Presse, médias

The Herald, Juillet 2008, dit à propos de la création de *Speakings* de Jonathan Harvey. *« It opens with a scream and ends in a hymn. Jonathan Harvey's orchestral work, Speakings, is possibly one of the most adventurous orchestral compositions ever undertaken by any composer, certainly in the recent history of music.*

It is almost outrageous in what it strives to depict and achieve. Even the gentle, wise and soft-spoken Harvey, among the most respected and revered of British composers, and a man given wholly to understatement, considers that Speakings is "the most complicated and ambitious composition I have ever written »

Une grande partie de l'orchestration de cette pièce est dérivée directement de solutions proposées par le moteur d'orchestration *Orchidée*.

3.3.2 Evénements publics

Les travaux de l'équipe IMTR et CataRT ont été présentés au festival « Earzoom » à la Galérie Kapelica, Ljubljana, Slovénie, 28.5.2009.

L'installation « Grainstick » liée au projet européen SAME et intégrant la synthèse par corpus a été présentée et ouverte au public pendant le festival Agora du 16 au 19.6.2009 et a attiré de nombreux visiteurs.

L'installation « Xe-Rocks » était présenté en vernissage à l'espace culture et multimédia Gantner, Bourogne, Territoire de Belfort le 27.6.2009 et est ouverte au public du 27.6. au 26.8.2009.

3.4 Valorisation industrielle

En raison du caractère confidentiel des informations relatives à ce paragraphe, celles-ci sont regroupées dans une annexe séparée dont la diffusion est restreinte aux destinataires concernés.

4 Annexe : thèses issues du projet

4.1 Thèse de Damien Tardieu

Cette thèse traite de la conception d'une nouvelle méthode d'orchestration assistée par ordinateur. L'orchestration est considérée ici comme l'art de manipuler le timbre d'un orchestre par l'assemblage des timbres des différents instruments. Nous avons proposé la formulation suivante du problème de l'orchestration assistée par ordinateur trouver les combinaisons de sons instrumentaux dont le timbre se rapproche le plus possible d'un timbre cible fournit par le compositeur. Cette cible est spécifiée sous forme de son enregistré. Les connaissances du système sur le timbre des instruments doivent être extraites de bases d'échantillons. Cette formulation pose un ensemble de questions liées à la description et à la modélisation des sons musicaux, à la perception de ces sons et au contexte particulier qu'est l'orchestre. La discussion de ces questions nous a conduit à considérer l'orchestration assistée par ordinateur comme un problème d'optimisation multicritères probabiliste où les critères sont les différents descripteurs du son.

La première étape est donc de définir un ensemble de descripteurs du signal sonore qui permette de décrire le timbre des instruments de l'orchestre. Ces descripteurs doivent être liés à la perception du timbre. Cet ensemble permet de décrire le contenu sinusoïdal du son, la forme du résiduel non-sinusoïdal, l'enveloppe spectrale, la modulation et la forme temporelle de l'énergie. Pour chaque descripteur, nous avons proposé une formule d'addition qui permet d'estimer le descripteur d'une combinaison à partir de ceux des sons qui la composent. Ces formules sont déduites de résultats d'acoustique des salles ou de la perception des sons. Une expérience de psychologie expérimentale sur la perception des combinaisons de sons a ainsi été réalisée à l'Université de McGill au Canada avec la collaboration du Professeur Stephen McAdams.

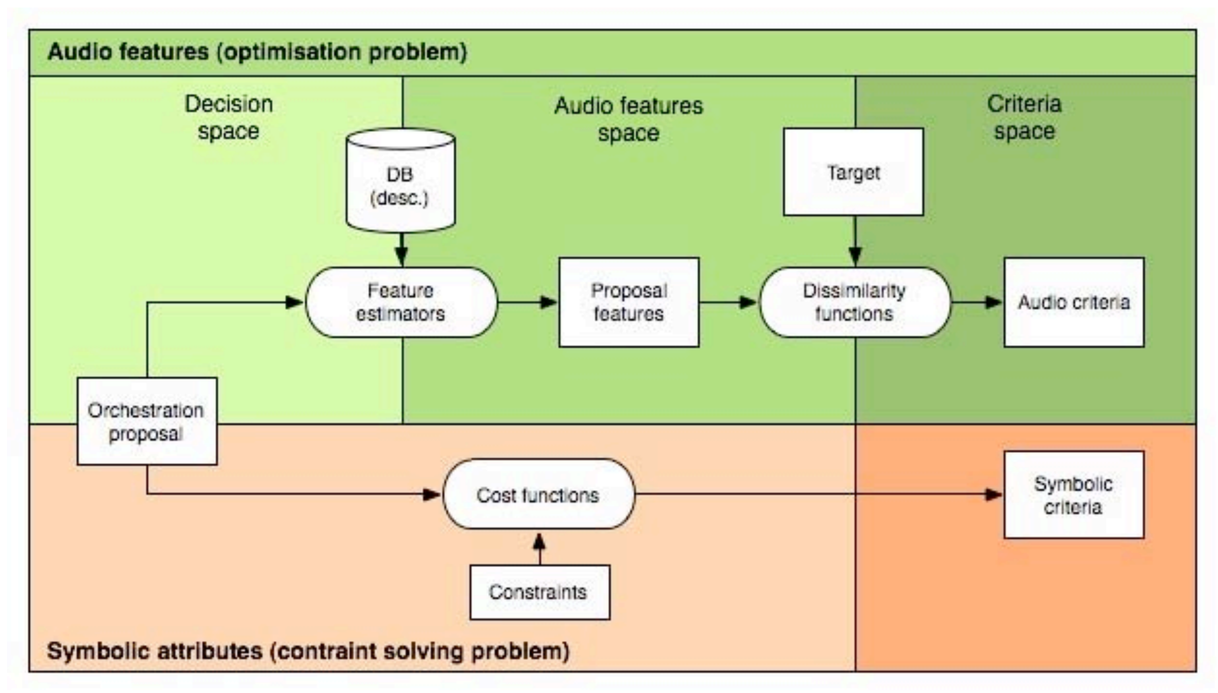
La seconde étape a consisté à définir une méthode d'apprentissage de ces descripteurs pour les différents instruments. Dans ce but, nous avons proposé une taxinomie des modes de jeu instrumentaux. L'apprentissage consiste alors à estimer la distribution des descripteurs pour les différentes valeurs des variables de la taxinomie. Ensuite une méthode d'agrégation des distributions permet de déduire des modèles pour un vaste ensemble de sons. Cette méthode d'agrégation, issue du domaine de la décision collective, permet un haut niveau de généralisation et donc la déduction de modèles de sons absents de la base d'apprentissage. À partir des modèles d'instruments, nous avons pu déduire le modèle d'une combinaison de sons instrumentaux et donc évaluer la probabilité que cette combinaison soit perceptivement proche de la cible.

Finalement les solutions proposées ont été testées par des tâches de classification d'échantillons solo et polyphoniques. Les résultats obtenus montrent les capacités du système à retrouver les sons composant un mélange.

Ces méthodes sont intégrées dans le moteur d'orchestration réalisé par Grégoire Carpentier qui à traité l'aspect optimisation du problème. Des versions intermédiaires de ce programme ont été utilisées dans plusieurs productions de compositeurs de L'IRCAM.

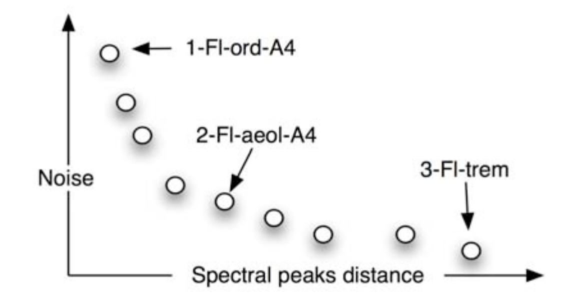
4.2 Thèse de Grégoire Carpentier

De toutes les composantes de l'écriture musicale, l'orchestration — ou l'art d'assembler les timbres intrumentaux — est longtemps demeurée, dans son enseignement comme dans sa pratique, une activité empirique. La difficulté de formaliser de manière rigoureuse l'ensemble des techniques inhérentes à cette discipline fait qu'aujourd'hui encore, l'orchestration resteun domaine peu abordé par l'informatique musicale et l'aide à la composition. Les rares outils actuels ramènent le problème de l'orchestration à la découverte, au sein de banques d'échantillons sonores instrumentaux, de combinaisons approchant au mieux un timbre fixé par le compositeur. Cette approche a été la nôtre. Mais là où les méthodes actuelles contournent systématiquement le problème combinatoire de l'orchestration par le recours à des principes de décomposition ou à des algorithmes de matching pursuit, l'originalité de notre démarche a été de placer les enjeux combinatoires au cœur de nos travaux et de traiter l'orchestration à la mesure de sa complexité.

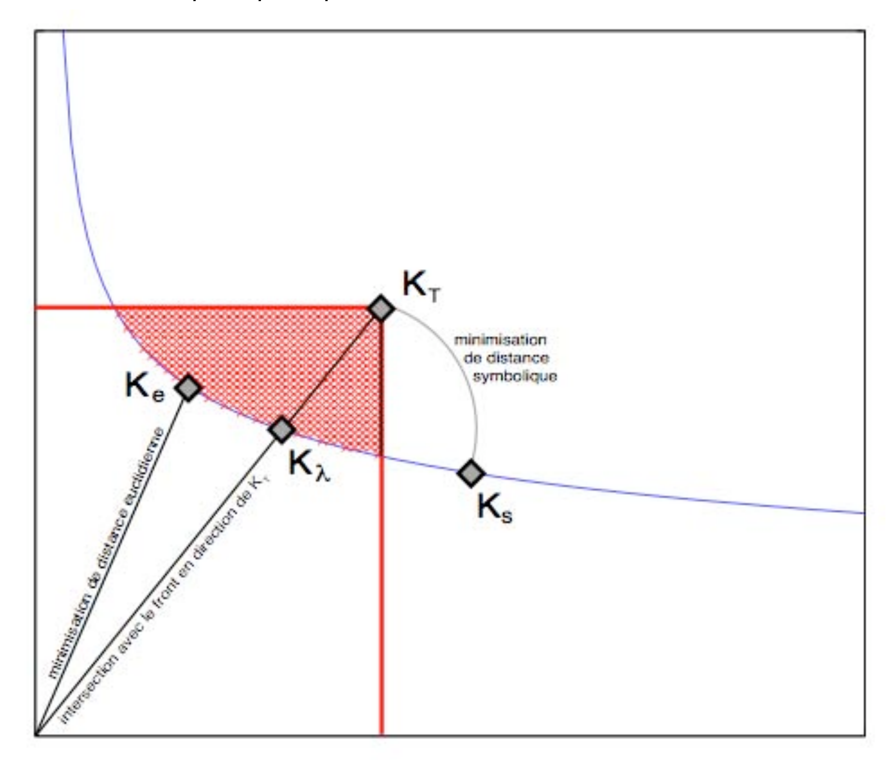


Espace de décision (connaisance instrumentale), espace de descripteurs (audio features), espace de critères (distances relatives sur les descripteurs)

Envisageant tout d'abord la question comme un problème de sac à dos multi-ob jectifs, nous avons montré que les non-linéarités dans les modèles de perception du timbre imposent un cadre théorique plus large pour l'aide à l'orchestration. Nous avons proposé une formalisation générique et extensible en nous plaçant dans un cadre de recherche combinatoire multicritère sous contraintes, dans lequel plusieurs dimensions perceptives sont optimisées conjointement pour approcher un timbre cible défini par le compositeur. Nous avons validé dans un premier temps notre approche théorique en montrant, sur un ensemble de problèmes de petite taille et pour une caractérisation exclusivement spectrale du timbre, que les solutions du problème formel (front de Pareto) correspondent à des propositions d'orchestration pertinentes.

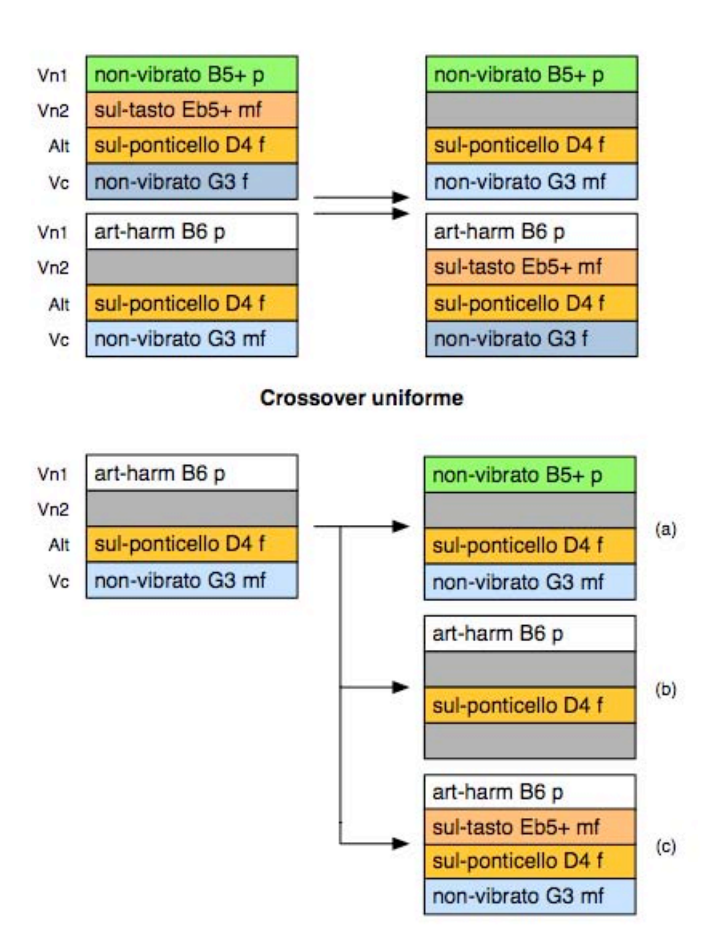


Espace perceptif à 2 critères et front de Pareto



Trois heuristiques pour trouver des solutions pertinentes dans lefront de Pareto lorsque ce dernier ne contient pas la solution théorique. Kλ(heuristiqueHλ) est obtenue par classement du front selon la norme de Tchebycheff induite; Ke (heuristiqueHe) par minimisation de la norme euclidienne, et Ks (heuristique Hs) par minimisationde la distance symbolique

Nous avons présenté alors un algorithme évolutionnaire permettant de découvrir en un temps raisonnable un ensemble de solutions optimales. S'appuyant sur la prédiction des propriétés acoustiques des alliages instrumentaux, cette méthode propose des solutions d'orchestration en fonction de critères perceptifs et encourage ainsi la découverte de mélanges de timbres auxquels le savoir et l'expérience n'auraient pas nécessairement conduit. En outre, la recherche peut-être à tout moment orientée dans une direction privilégiée.
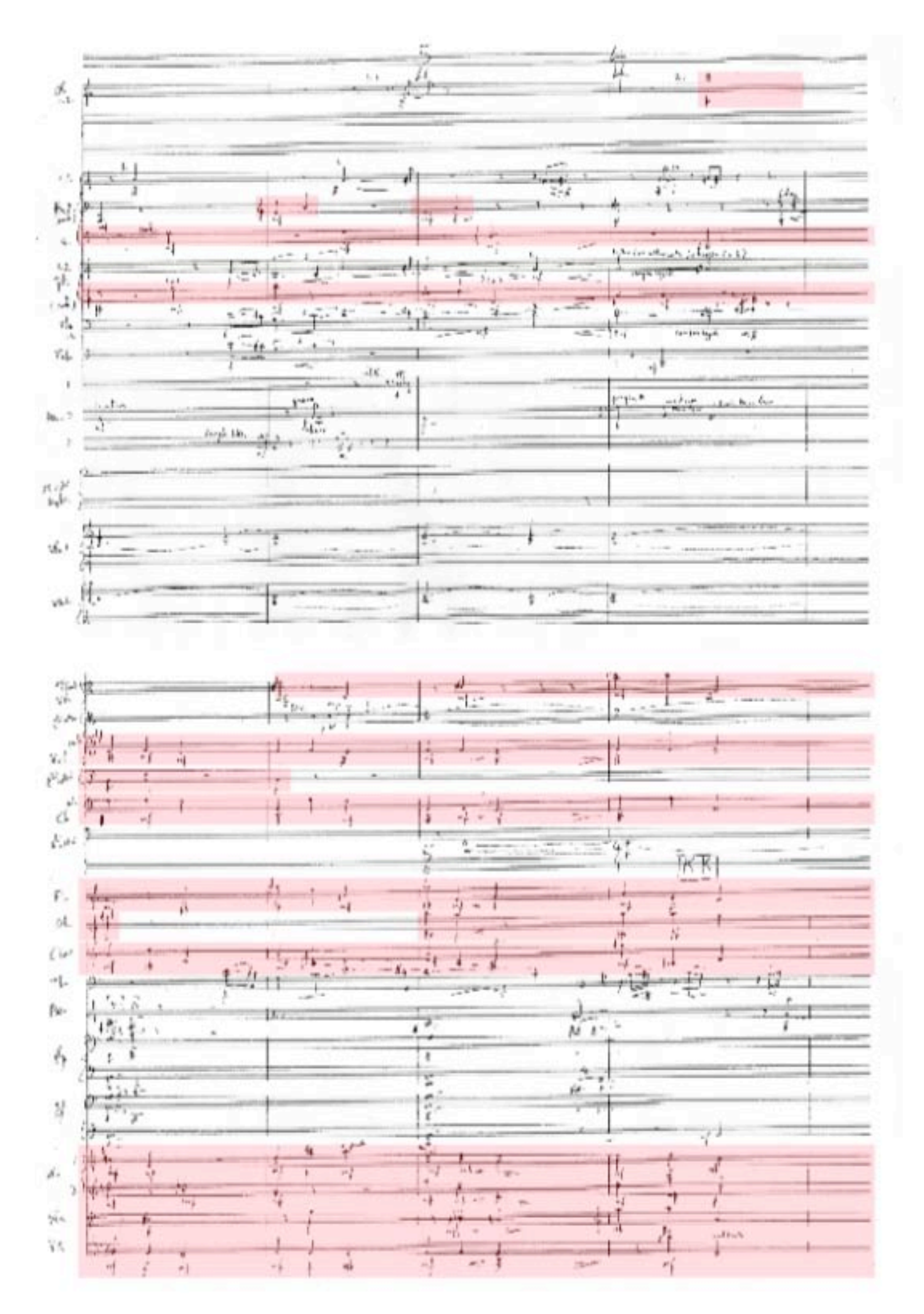


Types de mutation dans une population de mixtures instrumentales

	monophonique contraint		*monophonique libre*	
	v.1.0	random	v.1.0	random
Supériorité	**18.40 %**	2.04 %	**8.79 %**	1.23 %
Pseudo-dominance	**84.25 %**	15.75 %	**61.15 %**	38.85 %
Convergence	**83.44 %**	16.56 %	**78.73 %**	21.27 %
Diversité	69.12 %	**84.25 %**	25.15 %	**78.32 %**
	polyphonique contraint		*polyphonique libre*	
	v.1.0	random	v.1.0	random
Supériorité	**56.00 %**	0.00 %	**27.00 %**	0.00 %
Pseudo-dominance	**98.40 %**	1.60 %	**89.40 %**	10.60 %
Convergence	**95.20 %**	4.80 %	**98.00 %**	2.00 %
Diversité	54.60 %	**82.60 %**	40.60 %	**60.20 %**

Performances d'un algorithme génétique élémentaire par rapport à une recherche aléatoire.

Parallèlement, nous avons défini un cadre formel pour l'expression de contraintes globales et introduit une métaheuristique innovante de résolution, permettant de guider la recherche vers des orchestrations satisfaisant un ensemble de propriétés symboliques en lien direct avec l'écriture musicale.

Nous avons présenté enfin deux prototypes expérimentaux et une application finale d'aide à l'orchestration utilisables directement par les compositeurs, dans lequel l'exploration des possibilités de timbres est facilitée à travers une représentation multi-points de vue des solutions et un mécanisme interactif des préférences d'écoute.



Extrait de la partition de *Speakings* de Jonathan Harvey (Création Royal Albert Hall Aout 2008 BBC Symphony). Les parties en rose sont directement transcrites du moteur d'aide à l'orchestration Orchidée.